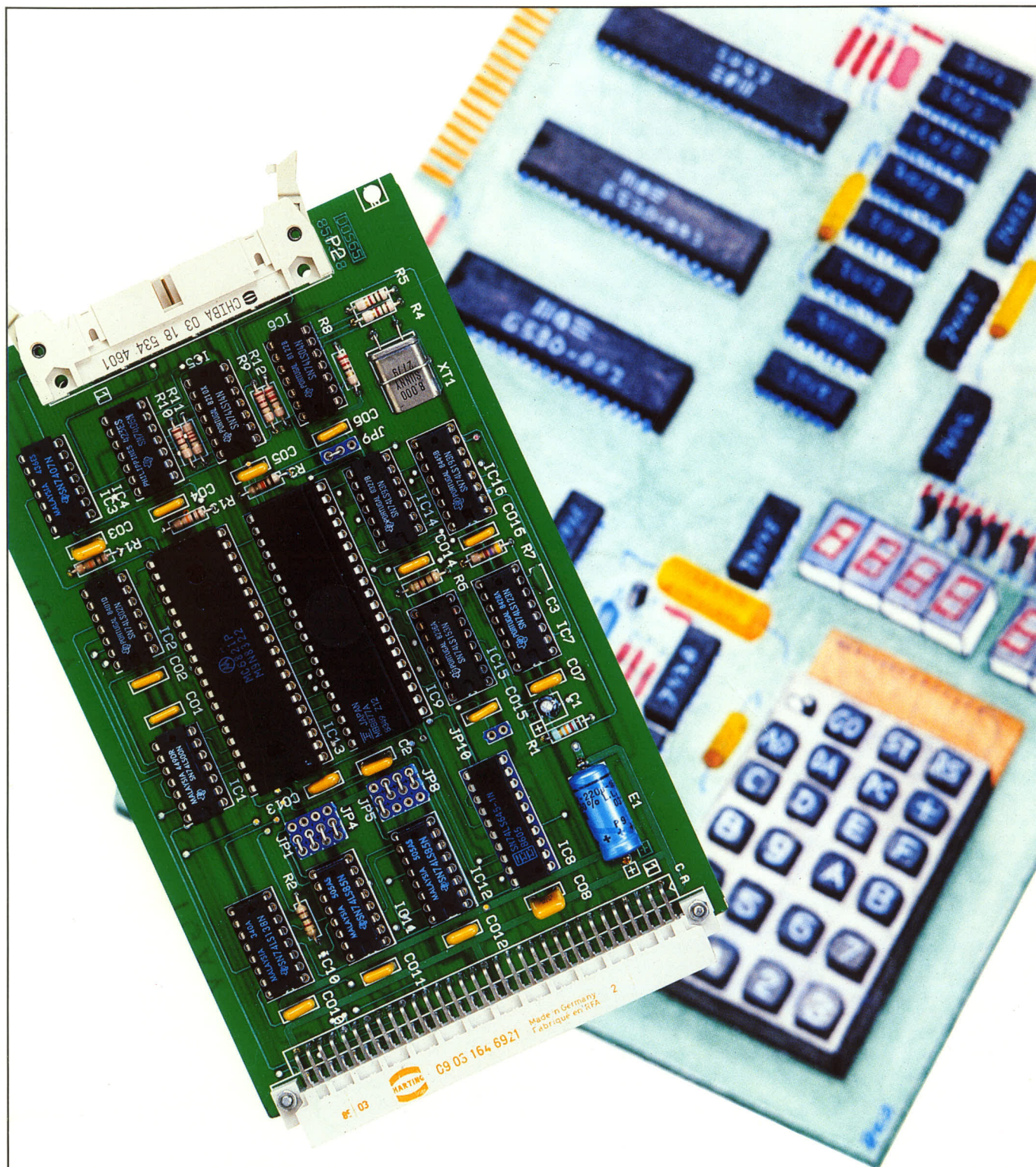


Twaalfde jaargang nr. 5 oktober 1988



DE 6502 KENNER

Vereniging

INFORMATIE.

De 6502 Kenner is een uitgave van de KIM Gebruikersclub Nederland. Deze vereniging is volledig onafhankelijk, is statutair opgericht en ingeschreven bij de Kamer van Koophandel en Fabrieken voor Hollands Noorderkwartier te Alkmaar, onder nummer 634305.

Het doel van de vereniging is het bevorderen van de kennisuitwisseling tussen gebruikers van computers die zijn opgebouwd rond een microprocessor uit de 6500-familie. Voorbeelden hiervan zijn onder andere: Elektuur EC-65, Commodore 64, Apple][, Elektuur Junior, Atari 600 en 800.

De eerder genoemde kennisuitwisseling komt onder andere tot stand door 6 maal per jaar de 6502 Kenner te publiceren, door de organisatie van landelijke bijeenkomsten voor de leden, het instandhouden van een softwarebibliotheek op cassette, floppy disk en papier en het beschikbaar stellen van een Bulletin Board.

Landelijke bijeenkomsten:

Deze worden gehouden op bij voorkeur de derde zaterdag van de maanden januari, maart, mei, september en november. De exacte plaats en datum worden steeds in de 6502 Kenner bekend gemaakt in de rubriek Uitnodiging.

Bulletin Board:

Voor het uitwisselen van mededelingen, het stellen en beantwoorden van vragen en de verspreiding van software wordt er door de vereniging een Bulletin Board beschikbaar gesteld. Dit Bulletin Board valt onder de verantwoordelijkheid van één van de bestuursleden en wordt bediend door een zgn. Sysop.

Software Bibliotheek:

Voor het beheer van de Software Bibliotheek streeft het bestuur er naar zgn. Software Coördinatoren te benoemen. Hierbij wordt gedacht aan een drietal coördinatoren; één voor EC-65(K) en Junior met OHIO DOS-65D, één voor DOS-65 en één voor diverse andere systemen zoals onder andere Elektuur Junior.

Het Bestuur:

Het bestuur van de vereniging wordt gevormd door een dagelijks bestuur bestaande uit een voorzitter, een secretaris en een

penningmeester en een viertal gewone leden.

Voorzitter:

Rinus Vleesch Dubois
Emiliano Zapataplein 2
2033 CB HAARLEM
Telefoon 023-330993

Secretaris:

Gert Klein
Diedenweg 119
6706 CM WAGENINGEN
Telefoon 08370-23646

Penningmeester:

John van Sprang
Tulp 71
2925 EW KRIMPEN A/D IJSSEL

Leden:

Adri Hankel (Bulletin Board)
Willem Kloosstraat 32
7606 BB ALMELO
Telefoon 05490-51151

Erwin Visschedijk
Dillelaan 11
7641 CX WIERDEN
Telefoon: 05496-76764

Gert van Opbroek (Redactie 6502 Kenner)
Bateweg 60
2481 AN WOUBRUGGE
Telefoon 01729-8636

Nico de Vries
Mari Andriessenrade 49
2907 MA CAPELLE A/D IJSSEL
Telefoon 010-4517154

Ereleden:

Naast het bestuur zijn er een aantal ereleden, die zich in het verleden bijzonder verdienstelijk voor de club hebben gemaakt:

Erevoorzitter:

Siep de Vries

Ereleden:

Mevr. H. de Vries van der Winden
Anton Mueller

=====

De 6502 Kenner:

De 6502 Kenner wordt bij verschijnen gratis toegezonden aan alle leden van de KIM Gebruikersclub Nederland. De kopij voor het blad dient bij voorkeur van de leden afkomstig te zijn. Alle kopij wordt door de redactie op bruikbaarheid en publicatiewaarde beoordeeld. Deze twee criteria, in samenhang met de actualiteit, bepalen of en zo ja wanneer het stuk gepubliceerd wordt. De redactie streeft er naar de kopij zoveel mogelijk in zijn oorspronkelijke vorm te plaatsen, Nederlandstalige kopij wordt daarom in principe niet naar een andere taal vertaald. De redactie streeft er naar een Nederlandstalig blad te maken doch het staat de auteur vrij een artikel geheel of gedeeltelijk in een andere taal te schrijven. Helaas kan de redactie, noch het bestuur, enige aansprakelijkheid aanvaarden voor de toepassing(en) van de gepubliceerde kopij.

Verschijningsdata:

De 6502 Kenner verschijnt op de derde zaterdag van de maanden februari, april, juni, augustus, oktober en december.

Redactie.

De redactie wordt momenteel gevormd door:
Gert van Opbroek

Correspondenten:
Jacques Banser (Sysop)
Bram de Bruïne
Antoine Megens
Nico de Vries
Rinus Vleesch Dubois

Redactieadres:
Gert van Opbroek
Bateweg 60
2481 AN Woubrugge

INHOUDSOPGAVE

Vereniging:

Informatie	2
Redactioneel	4
Uitnodiging clubbijeenkomst	5
Agenda voor de algemene ledenvergadering ..	6
Huishoudelijk Reglement	7

Algemeen:

Getallen deel 1	34
-----------------------	----

DOS-65:

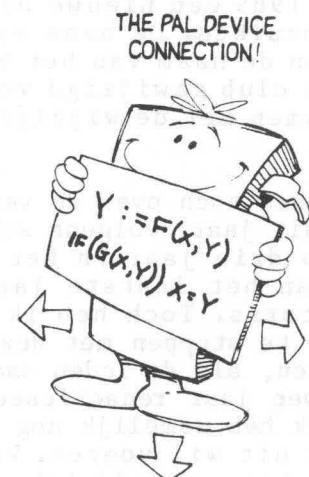
Viditel-65 binnenste buiten	10
De Eprommer: levering helaas vertraagd ...	18
IRQREST.MAC	19
Handigheidjes en foutjes in utilities	26
DOS-65 Game Library	40

Marktinfo:

Programmeerbare logica, wat is dat?	20
---	----

MS-DOS:

Van Micro Ade naar MASM	51
-------------------------------	----



Redactioneel

In het oktobernummer staat altijd de uitnodiging voor de ledenvergadering van november. Ook dit jaar is dat zo. Ik hoop van harte dat de ledenvergadering toch wat rustiger verloopt als vorig jaar. Mensen die vorig jaar al lid waren, zullen zich nog wel kunnen herinneren wat er allemaal gebeurd is. Ik hoop van ganser harte dat dergelijke zaken voorlopig niet meer voor zullen komen. Toch wordt de komende ledenvergadering voor de club heel belangrijk. Ik denk zelfs dat het voortbestaan van de club afhangt van de komende ledenvergadering. Het bestuur wil namelijk een ingrijpende beleidswijziging voorstellen. We willen, als club, processor-onafhankelijk worden. Om dat te mogen, hebben we de toestemming van de leden nodig. In principe moeten zelfs de statuten gewijzigd worden. Dat gaan we niet nu doen, maar als de leden de beleidswijziging ook zien zitten gaan we daar volgend jaar wel aan werken.

Als onderdeel van het nieuwe beleid moet ook de naam van het blad gewijzigd worden. Tenslotte suggereert de naam dat de lezers alles van de 6502 afweten en dat hoeft niet. Ik zie het als de taak van de redactie in november met een voorstel te komen. Ik denk daar al een tijdje over na maar heb nog niets echt pakkends gevonden. Misschien dat er mensen zijn die wel een goed idee hebben, laat die mij dan eens helpen. Ik ben namelijk van plan per januari 1989 een nieuwe naam voor het blad te gebruiken. De naam van de club zie ik los van de naam van het blad. Als de naam van de club gewijzigd wordt, dan gebeurt dit samen met de wijziging van de statuten.

Nu ik het toch over de vergadering heb, ik moet dit jaar, volgens schema aftreden. Ik heb nu drie jaar in het bestuur gezeten waarvan het laatste jaar als redactiesecretaris. Toch heb ik nog niet de behoefte te stoppen met deze functie. Ik heb besloten, als de leden daar mee instemmen, nog twee jaar redactiesecretaris te blijven. Ik heb namelijk nog een aantal ideeën die ik uit wil voeren. Vandaar dat ik me herkiesbaar gesteld heb.

Twee van de ideeën zult u in dit nummer al tegenkomen. Ik ben begonnen met een serie over getallen en ik heb de rubriek MS-DOS Corner gestart. Van het eerste heb ik een beetje verstand, van het tweede helemaal niet. Toch denk ik dat het zinvol is een vaste plaats voor MS-DOS in te ruimen. Ik hoop dat u mee wilt werken deze rubriek te

vullen. Natuurlijk hebben de leden met een Atari of wat dan ook ook recht op een eigen rubriek. Zo gauw er kopij voor deze systemen aanwezig is, wordt deze rubriek gestart. Daar MS-DOS tot voor kort binnen de club gediscrimineerd werd, ben ik van mening dat we wel iets extra's voor deze systemen mogen doen, vandaar de MS-DOS Corner in dit nummer.

Verder vindt ik het jammer dat er in dit nummer geen bijdrage over hogere programmeertalen is. Dit wordt echter veroorzaakt door het feit dat mijn floppies zo langzamerhand uitgeput zijn en ik eigenlijk ook de tijd niet heb speciaal voor een aflevering van de 6502 Kenner een programma te schrijven. Bovendien had ik geen inspiratie. Misschien de volgende keer beter.

Van Antoine Megens wordt in dit nummer deel 2 van de game library geplaatst. De gepubliceerde versie is in assembler. De auteur heeft echter niet stil gezeten en de game library ook in DOS-65 small C vertaald. Ook bouncing babies is ook in deze C beschikbaar. Al deze programmatuur staat op het Bulletin Board dat u bijna 24 uur per dag kunt bereiken onder nummer:

053-303902

Ook voor mensen met andere systemen is het bulletin board interessant. Er zijn o.a. file-areas voor Atari, Amiga, MS-DOS en speciaal voor communicatie. Het is zeker de moeite waard eens rond te neuzen en laat, als u ingelogd bent eens een berichtje voor de sysop Jacques Banser of uw redacteur Gert van Opbroek achter. Ik kijk meestal eens per week even wat er op het BBS staat zodat ik uw bericht zeker zal lezen.

Tenslotte nog een vraag. Ik denk dat het goed is, dat er iemand komt die zich actief bezig gaat houden met ledenwerving. Ik ben namelijk van mening dat het mogelijk moet zijn door wat meer aandacht aan de ledenwerving te besteden het ledental weer wat op te vijzelen. Zeker als we ook de andere systemen in de club binnenhalen, ligt er een groot potentieel aan leden. Wie wil zich daar eens mee bezig gaan houden? Indien u interesse hebt, kunt u contact opnemen met een van de bestuursleden. Tenslotte moet het toch mogelijk zijn pakweg 500 mensen voor onze club en of ons blad te interesseren.

Verder veel plezier met uw hobby en eventueel tot ziens in Almelo.

Gert van Opbroek

Uitnodiging voor de clubbijeenkomst

Datum: Zaterdag 19 november 1988
Locatie: Wijkcentrum 't Veurbook
Jan Tooropstraat 27
7606 Almelo
Tel.: 05490 - 10353

Bijz.: Algemene ledenvergadering
Toegang: Gratis

Routebeschrijving

Van uit het westen en het zuiden (A1/A35):
1. Aan het einde van de snelweg rechtsaf. Bij het eerstvolgende kruispunt MET VERKEERSLICHTEN linksaf, richting Wierden/Zwolle. Bij de eerstvolgende verkeerslichten rechtdoor. Bij de volgende verkeerslichten (links BP tankstation en Opel garage Kamp) gaat u rechtsaf.
2. U rijdt nu op de Windmolenbroeksweg. Doorrijden tot over de brug, dan de eerste straat rechts. Dit is de W. van Konijnenburgstraat. Na plm. 50 meter rechtsaf. Dit is de Tooropstraat. Met de bocht mee naar links. Na plm. 50 meter aan de rechterkant: 't Veurbrook.

Van uit het noorden (via de N 36):

1. Bij de stoplichten rechtsaf, richting streekziekenhuis. U bevindt zich nu op de rondweg om Almelo. Deze weg blijven volgen tot u het BP tankstation ziet bij dit kruispunt linksaf. Zie verder punt 2.

Met openbaar vervoer:

Vanaf NS-station Almelo met de stadsbus naar de wijk Molenbroek. Uitstappen bij de halte Windmolenbroeksweg. Schuin tegenover de bushalte staat een wegwijzer, daarop staat ook 't Veurbrook vermeld.

Programma:

9:30 Zaal open met koffie
10:00 Opening en aanvang algemene ledenvergadering (agenda zie elders)
12:30 Lunch, consumpties tegen betaling
13:30 Forum en markt

Aansluitend het informele gedeelte met de mogelijkheid om andermans systemen te bewonderen en Public Domain software uit te wisselen. **Breng daarom ook uw eigen systeem mee.**

17:00 Sluiting



Agenda voor de algemene ledenvergadering.

Datum: 19 november 1988
Plaats: Almelo

- 1: Opening ledenvergadering
- 2: Notulen van de laatste ledenvergadering 14 november 1987
- 3: Doelstelling van de club en beleidsplan 1989+ (zie toelichting)
- 4: Concept-begroting 1989
- 5: Verkiezing kascontrole-commissie 1988 (één lid), zie toelichting
- 6: Verkiezing kascontrole-commissie 1989
- 7: Huishoudelijk Reglement (zie toelichting)
- 8: Komende en gaande bestuursleden:

a) Volgens rooster aftredend:

Gert Klein (secretaris)
Gert van Opbroek (redactie)
Nico de Vries

Deze personen zijn allen herkiesbaar

b) Tussentijds aftredend en niet herkiesbaar:

John van Sprang (Penningmeester)
Erwin Visschedijk

Voor de functie van penningmeester wordt Jacques Banser voorgedragen.

Kandidaten voor de vacature Visschedijk kunnen zich uiterlijk voor aanvang van de vergadering schriftelijk bij de voorzitter melden. Tegenkandidaten voor de overige functies dienen zich eveneens schriftelijk uiterlijk voor aanvang van de vergadering bij de voorzitter te melden. Deze aanmelding dient vergezeld te gaan van de voordracht van tenminste twee leden.

9: Rondvraag

10: Sluiting

Toelichting

ad. 3

Het bestuur wil de doelstelling van de club wijzigen. In nr 57 van de 6502 Kenner heeft u een voorstel voor deze nieuwe doelstelling kunnen lezen. De omschrijving van de doelstelling en de verdere invulling worden op de vergadering aan de leden voorgelegd.

ad. 5

In de kascontrolecommissie voor 1988 is een vacature ontstaan vanwege het feit dat Coen Boltjes zijn lidmaatschap aanvang 1988 niet verlengd heeft.

ad. 7

Het Huishoudelijk Reglement.

Zoals u misschien nog weet, is op de ledenvergadering van november vorig jaar een commissie in het leven geroepen, die een Huishoudelijk Reglement moest gaan opstellen. Er is inmiddels enige tijd verstreken, en de commissie is thans in de gelegenheid u bijgaand een concept te kunnen presenteren. Het bestaan van het Huishoudelijk Reglement wordt door de Statuten van de vereniging voorgeschreven. Een dergelijk Reglement is bedoeld om richtlijnen te verschaffen hoe bepaalde artikelen uit de Statuten in de praktijk moeten worden uitgevoerd. Een voorbeeld hiervan is het stemmen bij volmacht.

De bedoeling is, dit concept in stemming te brengen op de ledenvergadering van november aanstaande. In deze vergadering kunnen uiteraard ook wijzigingen worden voorgesteld en ingevoerd. Mocht u op- of aanmerkingen hebben op dit concept, schrijf dan eens op, en stuur ze naar Nico de Vries. De commissie zal dan bekijken of ze in het thans voorgestelde Reglement kunnen worden ingepast.

Indien de ledenvergadering accoord gaat met dit of een gewijzigd Reglement, dan wordt het onmiddellijk bekrachtigd, zodat de vereniging het jaar 1989 kan beginnen met een bij de Statuten behorende Huishoudelijk Reglement.

De Commissie Huishoudelijk Reglement.

KIM GEBRUIKERSCLUB NEDERLAND

HUISHOUELIJK REGLEMENT (CONCEPT)

ARTIKEL 1. Begripsomschrijvingen.

In dit reglement wordt verstaan onder:

"vereniging"	De KIM GEBRUIKERSCLUB NEDERLAND, opgericht op de twee en twintigste juni negentienhonderd acht en zeventig te Castricum.
"statuten"	De statuten van de vereniging.
"bestuur"	Het bestuur van de vereniging zoals genoemd in de statuten.
"leden"	De leden van de vereniging zoals omschreven in de statuten.
"ledenvergadering"	Hetgeen daaronder wordt verstaan in de statuten.

ARTIKEL 2. Doel en geldigheid.

1. Dit reglement bevat nadere, voor de leden van de vereniging bindende bepalingen ter uitvoering van de statuten van de vereniging zoals bedoeld in artikel 16 van die statuten.
2. Toetsing van dit reglement aan de statuten, zoals bedoeld in artikel 16 van de statuten is voorbehouden aan de ledenvergadering.

ARTIKEL 3. Toelating tot het lidmaatschap.

1. Indien enig natuurlijk persoon zich heeft gewend tot het bestuur met het verzoek als lid to de vereniging te worden toegelaten, zulks al dan niet gedaan hebbende op een wijze als bepaald in artikel 4 van de statuten, wordt hem het lidmaatschap niet dan schriftelijk en met opgave van reden geweigerd. Het bestuur draagt zorg dat zulk een weigering vergezeld gaat van een afschrift van artikel 4 van de statuten.
2. Een ieder die door het bestuur wordt toegelaten als lid, ontvangt door de zorg van het bestuur de statuten en het huishoudelijk reglement.
3. Het bestuur neemt een verzoek tot toelating tot het lidmaatschap niet in behandeling indien zulk een verzoek afkomstig is van enig persoon welke door de ledenvergadering is ontzet uit het lidmaatschapsrecht zoals bedoeld in artikel 8 lid 4 van de statuten.
4. Het bestuur is bevoegd te verlangen dat de schriftelijke wens tot toelating als bedoeld in artikel 4 van de statuten plaats heeft op een wijze en in een vorm zoals door het bestuur in het belang van een goede ledenadministratie wordt nodig geacht. In dat geval zal het bestuur zorg dragen dat de betrokken persoon in het bezit wordt gesteld van de daartoe benodigde administratieve middelen.

ARTIKEL 4. Toegang tot bijeenkomsten.

1. Toegang tot bijeenkomsten welke door de vereniging worden georganiseerd hebben:
 - A. Zij die daar toegang hebben op grond van artikel 6 van de statuten, en bovendien:
 - B. Zij die door het bestuur daar worden toegelaten.
2. Onverminderd het bepaalde in artikel 5 tweede lid van de statuten, worden alle leden in de gelegenheid gesteld aan een ledenvergadering deel te nemen zonder dat hiertoe enige geldelijke vergoeding aan de vereniging verschuldigd zal zijn.
3. Personen welke geen toegang tot de ledenvergadering hebben, worden desondanks door het bestuur toegelaten tot dat deel van die ledenvergadering waarin enig beroep dat zij op grond van de statuten op de ledenvergadering hebben gedaan, in die vergadering aan de orde zal zijn, behoudens in die gevallen waarin naar het oordeel van het bestuur de aanwezigheid van de betrokkene directe schade aan de vereniging kan berokkenen.

ARTIKEL 5. Ontzetting uit het lidmaatschap.

1. Onverminderd het bepaalde in artikel 8 van de statuten zal het bestuur van een ontzetting uit het lidmaatschap eveneens kennis geven aan de ledenvergadering,

DE 6502 KENNER

Vereniging

ongeacht of het betrokken lid al dan niet gebruik maakt van zijn recht op beroep op de ledenvergadering. Het bestuur draagt zorg dat het betrokken lid al of niet ten overvloede, wordt gewezen op zijn recht van beroep op de ledenvergadering.

2. Indien een lid hangende een procedure tot ontzetting uit het lidmaatschap te kennen geeft zijn lidmaatschap te willen beeindigen, wordt dat lid beschouwd als door het bestuur te zijn ontzet uit het lidmaatschap, onverminderd het recht van beroep op de ledenvergadering.
3. De kennisgeving als bedoeld in het eerste lid van dit artikel blijft achterwege indien de ontzetting uit het lidmaatschap uitsluitend werd gedaan wegens het niet voldoen van de verschuldigde contributie door een lid. Dezelfde ontzetting uit het lidmaatschap kan door het bestuur worden beschouwd indien het betrokken lid binnen naar het oordeel van het bestuur redelijke termijn alsnog aan zijn verplichtingen heeft voldaan.
4. Ontzetting uit het lidmaatschap van een lid dat deel uitmaakt van het bestuur dan wel deel uitmaakt van een commissie benoemd door de ledenvergadering, is voorbehouden aan de ledenvergadering.

ARTIKEL 6. Werkwijze van het bestuur.

1. Het bestuur draagt zorg dat de taken van tenminste de voorzitter, de secretaris en de penningmeester, rekening houdend met het bepaalde in artikel 10 van de statuten, schriftelijk en voldoende gedetailleerd zijn vastgelegd, zodanig dat duidelijkheid bestaat over de werkwijze van deze bestuursleden, hun middelen en bevoegdheden.
2. Het bestuur zal trachten te voorkomen dat de secretaris en de voorzitter gelijktijdig aftreden.
3. Het bestuur draagt zorg dat eigendommen van de vereniging als zodanig zijn geregistreerd, naar behoren zijn verzekerd tegen schade welke kan ontstaan door diefstal of brand of enig van buiten komend onheil en naar Nederlands recht zijn gevrijwaard van beslaglegging op grond van aanspraken door derden tegenover de personen die de goederen in beheer of bruikleen hebben.
4. Het bestuur organiseert tenminste vijf maal per verenigingsjaar een bijeenkomst van de leden en ten minste twee maal per jaar een ledenvergadering. Ledenvergaderingen kunnen worden gehouden op dezelfde dag en op de dezelfde locatie als een bijeenkomst.
5. Behoudens goedkeuring door de ledenvergadering en met inachtneming van de statuten en dit reglement, zal het bestuur bevoegd zijn al datgene te doen dat zij in het belang van de vereniging acht.

ARTIKEL 7. De redactie van het huisorgaan.

1. Het bestuur wijst uit haar midden tevens een bestuurslid aan dat uitsluitend is belast met de redactie en uitgifte van het huisorgaan van de vereniging. Deze bestuursfunctie draagt de naam "redactiesecretaris".
2. De redactiesecretaris benoemt een commissie van leden bestaande uit zoveel personen als door hem wordt wenselijk geacht om hem behulpzaam te zijn bij zijn taak. Deze commissie draagt de naam "redactiecommissie". De redactiecommissie wordt voorgezeten door de redactiesecretaris en komt zo vaak bijeen als zij in het belang van de kwaliteit van het huisorgaan noodzakelijk acht.
3. De redactiesecretaris als zodanig en de leden van de redactiecommissie zijn niet bevoegd de vereniging op enigerlij wijze te vertegenwoordigen, behoudens bevoegdheden door het bestuur hen verleend.
4. Onverminderd het gestelde in het eerste lid kan, indien zulks naar het oordeel van het bestuur in het belang van de vereniging is, het beheer van de bibliotheek zoals omschreven in artikel 10 derde lid onder A van de statuten worden opgedragen aan de redactiesecretaris, zulks met inachtneming van het terzake bepaalde in artikel 8 van dit reglement.

ARTIKEL 8. Beheer van de bibliotheek.

DE 6502 KENNER

Vereniging

1. Het bestuur draagt zorg dat aan de leden generlij geldelijke vergoeding wordt gevraagd ter verkrijging van verstrekkingen uit de bibliotheek. Hiervan uitgezonderd is een vergoeding welke slechts dient ter dekking van de daadwerkelijk ten behoeve van zulk een verstrekking ontstane kosten.
2. Onder "artikel" zoals genoemd in artikel 7 van de statuten dient te worden verstaan al datgene dat terzake van enig onderwerp aan de vereniging ter beschikking wordt gesteld, ongeacht de vraag of de inbrenger al dan niet rechten op het ingebrachte kan doen gelden.
3. Het bestuur zal opname van artikelen in de bibliotheek weigeren indien het haar bekend is dan wel een redelijk vermoeden bestaat dat met de inbreng van het artikel door de inbrenger enige handeling in strijd met de wet wordt verricht.
4. Onverminderd het bepaalde in het derde lid draagt de inbrenger van enig artikel alle verantwoordelijkheid voor de oorspronkelijkheid daarvan en vrijwaart hij de vereniging van elke aanspraak door derden terzake aantasting van enig bestaand octrooi, merknaam of auteursrecht.

ARTIKEL 9. Stemming bij volmacht.

1. Een lid kan zich ter ledenvergadering doen vertegenwoordigen door elk ander lid dat ter vergadering stemrecht bezit. Hij dient daartoe het lid dat hem zal vertegenwoordigen een schriftelijke volmacht te verlenen, welke tenminste moet bevatten:
 - A. Naam, voorletters, volledig adres en handtekening van het lid dat de volmacht wenst te verlenen.
 - B. Naam, voorletters, en volledig adres van de gemachtigde.
2. Een lid kan ter ledenvergadering niet meer dan een (1) ander lid vertegenwoordigen.
3. Een lid dat door een ander lid is gemachtigd stelt bij aanvang van de vergadering zijn volmacht ter hand aan het bestuur. Het bestuur maakt vervolgens alle aanwezige volmachten als "ingekomen stuk" aan de ledenvergadering bekend.
4. In geschillen betreffende de geldigheid van een volmacht, beslist de ledenvergadering. Indien deze beslissing een stemming vereist, vindt deze plaats als stemming over zaken.
5. Stemming bij volmacht is niet toegestaan terzake van wijziging van de statuten en/of ontbinding der vereniging.

ARTIKEL 10. Inwerkingtreding en wijziging van dit reglement.

1. Dit huishoudelijk reglement treedt in werking onmiddellijk na goedkeuring door de ledenvergadering en zal worden gepubliceerd in de eerst volgende editie van het huisorgaan waarvan op het moment van inwerkingtreding de redactie nog niet is afgesloten.
2. Wijziging van dit reglement behoeft de goedkeuring van de ledenvergadering bij gewone meerderheid van stemmen.

ARTIKEL 11. (Met opzet opengelaten).

N.B. Dit artikel wordt misschien benut om het bestaan en functioneren van een kascontrolecommissie te regelen.

ARTIKEL 12. Geschillen.

In alle gevallen waarin dit huishoudelijk reglement niet voorziet beslist het bestuur, behoudens goedkeuring door de ledenvergadering.

DE 6502 KENNER

DOS-65 Corner

```
*****
**      VIDITEL-65 BINNENSTE BUITEN      B. de Bruine      **
*****
```

I. De configuratiepagina

=====

Viditel kent een configuratiepagina (Viditel.cfg) die de waarden bevat om met ieder systeem volgens eigen wensen te kunnen werken. Hieronder staat de sourcelisting van deze file:

```
;* * * * *
;Program name: VICFG.mac
;Creation date: 12 feb 1988
;Update: Seperated from vidi.mac
;Version: V 3.0
;Programmer: Bram de Bruine
;Hardware: DOS 65
;Purpose: Configurationvariables viditel-65
;Program description: Assemble this file with:
;                               AS -bVIDITEL.cfg VICFG
;* * * * *
;
;
;-Jumptable viditel-
1000 VIDI EQU $1000 ;Start address of viditel-65
1003 SEN2 EQU $1003 ;Send A to modem, x destroyed
1006 PARPROG EQU $1006 ;Program acia and modem
1009 IGET EQU $1009 ;Get key from keyboard
100C LASTIME EQU $100C ;Display time in statusline
100F SHAST EQU $100F ;Transmit string to modem
1012 JMAIN EQU $1012 ;Jump common entry mainmenu
1015 JVITXT EQU $1015 ;Printstring in viditel format
1018 JBREAK EQU $1018 ;Test if breakkey is pressed

0700 ORG $0700 ;configuration page $700-$7ff

0700 CFG ;Hayes-strings
0700 0000000000 HAYSTR FCC 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0 ;Y=00
0710 0000000000 FCC 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0 ;Y=10
0720 0000000000 FCC 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0 ;Y=20
0730 0000000000 FCC 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0 ;y=30
0738 4154413B0D FCC 'ATA;\r',0,0,0,0 ;y=38

;Macro strings
0740 4B4C414E54 MC1 FCC 'KLANTNUMMER_',0,0,0,0 ;Y=40
0750 434F44454E MC2 FCC 'CODENR.',0,0,0,0,0,0,0,0 ;Y=50
0760 5052495645 MC3 FCC 'PRIVECODE_',0,0,0,0,0,0,0,0 ;Y=60
0770 4D616E792D MC4 FCC 'Many-used-string' ;Y=70
0780 0D00000000 FCC '\r',0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
;Notice: the last macro has double length !

;Crtctab viditel screens
0790 7E 28 50 VIDTAB FCB $7E,$28,$50,$88 ;40 Char
0793 88
0794 1E 00 19 FCB $1E,$00,$19,$1C ;24 lines
0797 1C
0798 00 09 60 FCB $00,$09,$60,$09 ;9 scanlines, blinking cursor
```


DE 6502KENNER

DOS-65 Corner

```

079B 09
079C 00 00 00      FCB      $00,$00,$00,$00 ;Not used
079F 00

;Crtctab used by dos-65
07A0 7E 50 5F      DOSTAB  FCB      $7E,$50,$5F,$88 ;80 char
07A3 88
07A4 1E 05 19      FCB      $1E,$05,$19,$1C ;24 lines
07A7 1C
07AB 80 09 00      FCB      $80,$09,$00,$09 ;9 scanlines, block cursor
07AB 09
07AC 00 00 00      FCB      $00,$00,$00,$00 ;Not used
07AF 00

07B0 20202020      RES      4,$20      ;Futural expansion
07B4 415444      ATD      FCC      'ATD'      ;Hayes dial command
07B7 20      PUTD      FCC      ' '      ;Pulse, tone or def dialing
07B8 0000000000    DIALBUF  RES      17,0      ;Buffer used by autodial

;Configuration flags
07C9 02      CLOCK      RES      1,2      ;Processorspeed 1,2,3 & 4 mhz
07CA FF      SPLIT      RES      1,$FF      ;Split baud modem 75/1200
07CB 00      HAYES      RES      1,0      ;Zero, if smart modem used
07CC 00      EXTRA      RES      1,0      ;Autobaud modem initialisation
07CD E1      RES      1,$E1      ;In use for astrid (fifo)
07CE 02      PROTOCO      RES      1,2      ;Protocol=videotex (fixed)
07CF 00      TIMFL      RES      1,0      ;0=update time, <0> don't

;*:~*:~*:~* do not change sequence table below :~*:~*:~*
07D0 FF      ECHO      FCB      $FF      ;Full duplex = ff {default}
07D1 00      ANSORG      FCB      $0      ;Originate = 00 {default}
07D2 42      WORDFOT      FCC      'B'      ;A=8N1, B=7E1 {default}
07D3 04      BAUDPAR      FCB      4      ;Baudrate {default}
;1=2400/2400
;2=1200/1200
;3= 300/300
;4=1200/75
;5=dos specified
;*:~*:~*:~* end table :~*:~*:~*
07D4 FF      CFG1      FCB      $FF      ;0=astrid, $ff=viditel
07D5 33      CFG2      FCC      '3'      ;Version number hi
07D6 FF      EXTOSC      RES      1,$FF      ;00=external acia osc
;Ff=via-osc for acia (only if split=0)
07D7 32      CHRGEN      RES      1,'2'
; '0'=octopus: (8 sc lines, no d65 conversion)
; '1'=chrg1: 8 sc lines, 4 reserved.
; '2'=chrg2: 9 sc lines (lowest extend) 4 res.
; '3'=chrg3: 9 sc lines (middle extend) 4 res.

07D8 05      MONIT      RES      1,5      ;5=50hz,6=60hz,7=71hz
07D9 FF      AUTREV      RES      1,$FF      ;Auto reveal on/off
07DA FF      BANSER      RES      1,$FF      ;00=bansoft control i/o
;ff=no controllines of via

07DB 01      PARCHK      RES      1,01
;Mask parity check: 00 no check
; 01 check parity
;Keycodes menu transfer parameters/dialer/hayesmacro's
07DC 0B      KEYUP      FCB      $0B      ;^
07DD 0A      KEYDOW      FCB      $0A      ;Lf
07DE 08      KEYBAC      FCB      $08      ;Bs
07DF 09      KEYFOR      FCB      $09      ;Ht

```

```
07E0 1D      KEYBRK  FCB      '\E'+2      ;Common breakkey ^]
07E1 17      KEYPUP  FCB      $17          ;^w page up
07E2 1A      KEYPDO  FCB      $1A          ;^z page down
07E3 0D      USERKEY FCB      $0D          ;^U = ^M viditel mode
07E4 0000     RES      2,0              ;Future expansion
```

```
                ;Keycodes macro expansion
07E6 0E      KEYMAC  FCB      $0E          ;^n macro-key (2nd)
07E7 20 1510  MKHELP JSR      JVITXT
07EA 31      KEYMAC1 FCC      '1'          ;Macro 1
07EB 2C      FCC      ' '
07EC 32      KEYMAC2 FCC      '2'          ;Macro 2
07ED 2C      FCC      ' '
07EE 33      KEYMAC3 FCC      '3'          ;Macro 3
07EF 2C      FCC      ' '
07F0 34      KEYMAC4 FCC      '4'          ;Macro 4
07F1 00      FCB      0
07F2 60      RTS
```

END

Errors detected: 0

List 1: Sourcelisting Vicfg.mac

De configuratiepagina is als volgt opgebouwd:

700-78F: Macro's,
 790-79F: Crtc-tabel viditel (40 karakters),
 7A0-7AF: Crtc-tabel dos-65 (80 karakters),
 7B0-7DB: Configuratievariabelen,
 7DC-7F2: Functietoetsen.

Men kan de configuratiefile genereren met AS of met MCONFIG. Mconfig stelt een aantal vragen die beantwoord moeten worden. Aan de hand van die antwoorden word viditel.cfg samengesteld. Ook met As kan men deze file maken, men moet dan in VICFG.mac de juiste waarden met ED invullen in de sourcelisting. (zie voorbeeld boven in list 1.)

Viditel word gestart vanuit een commandfile. In deze file moet een LOAD VIDITEL.cfg staan. Nadat deze file geladen is kan men toch nog wijzigingen aanbrengen in de configuratie door er een kleine file overheen te laden. Dit kan handig zijn als men een afwijkende monitor heeft, en dus de 6845 anders geprogrammeerd moet worden.

De commandfile VIDI kan er dan als volgt uitzien:

```
Load viditel.bin
Load viditel.cfg
Load IBMVDU.bin      of Load IBMVDU 790,7AF
```

Op dezelfde manier kan men macro's overschrijven. Dit kan vanuit de mainmode van viditel gebeuren met: !LOAD MACFIL
 Macfil is dan bijvoorbeeld een file die van 740-78f loopt, en bestaat uit asciittekst, afgesloten met een \$00-byte.

Het is mogelijk om de functietoetscodes te wijzigen. Dit gaat het gemakkelijkst met Mconfig. Zo kan men macro 1 aanroepen met ESC A om de A(cces)code te versturen als locatie \$7E6 een ESC (\$1B) bevat en er op adres \$7EA een 'A' (\$41) staat.

De eerste vijf macro's worden als Hayes codestrings gebruikt door

viditel-65, indien men de configuratievariabele HAYES actief geprogrammeerd heeft. Dit betekent dat men deze macro's moet vullen met informatie volgens de voorgeschreven regels. Heeft men geen hayesmodem, dan maakt men HAYES inactief door er \$FF te zetten. Viditel-65 gebruikt deze strings dan niet om het modem in een bepaalde toestand te brengen, en men kan ze vrij programmeren.

Het blok met systeemafhankelijke waarden (\$7B0-\$7DB) zorgt ervoor dat het programma opstart met een geschikte default baudrate, woordformaat, clock-frequentie van het systeem, enz, enz.

Een uitzondering is het buffer vanaf \$7B8. Hier komt het telefoonnummer te staan dat men selecteert in de autodialer. Het nummer wordt uit deze buffer gehaald voordat dat nummer daadwerkelijk gebeld wordt. Ook hier kan men dus door de 'overschrijf-truc' het te bellen nummer op allerlei manieren manipuleren.

Conclusie: Door een aparte pagina met systeemafhankelijke waarden te gebruiken ontstaat een zeer flexibel programma, dat door bijna iedereen aan te passen is aan zijn eigen systeem en zijn eigen wensen.

II. De Commandoset

=====

De meest recente versie van viditel is V3.0. Men kan controleren welke versie men heeft door de V(ersion)toets in te drukken. Men krijgt dan het versienummer, gevolgd door de geconfigureerde modemcode en de releasedatum geprint. Zo betekent V2.0 HI april 1987, dat men versie 2 heeft, dat er geconfigureerd is voor een Hayesmodem met interspeeder, (HI) en dat de versie vrijgegeven is in april 1987.

Met het "!"commando beland men in de dosmode. Hier kan men doscommando's geven, zoals SEE FILE. Men kan dit tijdelijk blokkeren/starten met ^S, of afbreken met ^Z. Een ^C heeft als FATALE gevolg dat men definitief terugkeert naar dos-65.

III. Het printen van viditelpagina's

=====

Vanaf versie 3 kent viditel-65 de mogelijkheid om een viditelpagina af te laten drukken door de printer. Het printprogramma dient door de commandfile vidi geladen te worden. Indien men dit niet doet en men geeft toch een ^P (van Print) dan zal de melding 'Printmodule not present' volgen. Vanuit de dosmode kan men dan het printprogramma alsnog laden met !load vipa.bin. De printmodules die standaard bijgeleverd worden (vipa.bin voor alle printers, en vipradix.bin voor de Star-rdx10 printers) zijn beide voorzien van een time-out mechanisme en een breakdetectie routine. Dit betekent dat als de printer niet klaar staat voor gebruik, er na ca. 10 seconden automatisch een melding komt 'Printer not ready' en er teruggekeerd wordt naar de viditel terminalmode. Meestal ziet men al sneller dat de printer niet aangesloten is, en daarom kan men ook direct ingrijpen met de breektoets. Vipra.bin initialiseert de printer niet, vipradix.bin doet dit wel, en alleen de eerste keer dat hij aangeroepen wordt. Wil men iets printen en de printer is niet ready, dan gaat de initialisatie (klaarzetten van de videotex grafische karakterset) mis. Remedie: !load vipradix.bin nog een keer, zodat de initialisatievlag weer gereset wordt, en alles correct verloopt. Deze truc kan men ook toepassen als men de printer even uitgezet heeft, en de karakterset opnieuw geprogrammeerd dient te worden.

IV. Van viditelformaat naar ascii

=====

Met de nieuwe utility VIPRINT is het mogelijk om een .vid file te listen op het scherm in asciiformaat. Met > -tf Demo.asc Viprint demo.vid wordt een

DE 6502 KENNER

DOS-65 Corner

videotextfile omgevormd tot een asciifile die met ED te bewerken is.

```

ALLCTR  AND    #$7F
        PHA
        LDA     IGNORE           ;Test if previous char is attr. prec.
        BNE     22.F
        MOVEQ   $FF,IGNORE       ;Set no attribute
        PLA
        JMP     1.B
22      PLA
        CMP     #'E'             ;Videotex attribute marker ?
        BNE     23.F
        MOVEQ   0,IGNORE         ;Set attribute preceeder detected
        LDA     #$20
        JMP     6.F
23      CMP     #$7F             ;Del is printed as ^?
        BNE     24.F
        LDA     #$20
        JMP     6.F
24      CMP     #$20             ;Control char ?
        BPL     6.F
        CMP     $FF              ;Form feed --> cls
        BNE     99.F
        JSR     out              ;Cls
99      CMP     #'r'             ;Return ?
        BNE     16.F
        PHA
        LDA     #0
        STA     COUNT           ;Reset count
        PLA
        JSR     out
        JMP     8.F
16      CMP     #'n'             ;lf ?
        BNE     14.F
        JMP     9.F

14      CMP     #HT              ;Cursor right ?
        BEQ     78.F
        CMP     #BS              ;Cursor left ?
        BNE     77.F
        LDA     COUNT
        BEQ     15.F
        DEC     COUNT           ;Reverse count
15      LDA     #BS
        JMP     77.F
6       INC     COUNT           ;Count char
7       JSR     out
8       LDA     COUNT
        CMP     #40
        BMI     1.B
9       LDA     #0              ;Reset counter
        STA     COUNT
        JSR     print           ;And new line
        FCC     '\r',0
        JMP     1.B
77      NOP
        JMP     7.B
78      NOP
        JMP     6.B
    
```

List 2: Hoofdgedeelte Viprint

Momenteel is Viprint nog een eenvoudig programmaatje dat om de veertig karakters een <CR> (= \$0D) invoegt en displayattributen eruitfiltert. Beter zou het zijn om ook nadat een 'set graphic' code ontvangen is alle navolgende grafische informatie te vervangen door een spatie.

V. Fout in Mconfig 3.0

Helaas is er een storende fout in Mconfig 3 geslopen. Er worden 2 bytes op de verkeerde plaats gezet, met als gevolg dat de VDU-kaart niet correct geprogrammeerd wordt, en men een onleesbaar beeldscherm krijgt. Dit is op te lossen door de Crtctabel voor 80 karakters over viditel.cfg heen te laden, zoals hiervoor in &I. beschreven, of door de verbeterde mconfigversie V3.1 te gebruiken.

VI. Viditel-65 versnellen

Viditel-65 wordt verspreid voor een standaard Nmos 6502 processor. Het is echter geschreven voor een Cmosprocessor met gebruik van de nieuwe instructies. Maar alleen die nieuwe instructies zijn gebruikt die eenvoudig vervangen kunnen worden door een macro in AS, die de code omvormt naar de standaardcode. Indien men een Cmosprocessor in zijn systeem heeft zitten, kan men de .bin file aanzienlijk verkleinen, waardoor de gebruikerssnelheid ook iets toeneemt.

AS -1 Vidi genereert 6502 code.

AS Vidi genereert 65C02 code.

De file CMOS.mac kan ook voor andere programma's van nut zijn. Men dient er wel rekening mee te houden dat men dan toch nog aanpassingen aan zijn programma moet aanbrengen, bijvoorbeeld omdat een branch 'out of range' raakt. Bij viditel is hier rekening mee gehouden, en kan probleemloos geassembleerd worden.

```

; * * * * *
;Program name: CMOS
;Creation date: 18 feb 1988
;Update:
;Version: V1.0
;Programmer: Bram de Bruine
;Hardware: DOS 65
;Purpose: Emulation 65c02 code on standard 6502
;Program description: Macro calls
; * * * * *

```

```

;This program emulate only the new mnemonics who does nt use
;the indirect addressing mode. The macros below use a lot
;more memoryspace then solutions in normal NMOS.

```

```

        argl    zero           ;Default: assemble for cmos processor
arg      argm    prosel        ;If argument is 1, then emulate

zero     equ     0
emulate  equ     1

        if      prosel==emulate ;set emulate mode

BRA      MACRO   ADR           ;BRANCH ALWAYS
        JMP      ADR

```



```

ENDM

STZ      MACRO      OP      ;STORE ZERO in memory abs
        PHA          ;but not yet: ABS,X - Z, Z,X
        LDA          #0
        STA          OP
        PLA
        ENDM

INCA      MACRO      ;INCREMENT A  C,V destroyed
        CLC
        ADC          #1      ;A:=A+1
        CLC
        ENDM

DECA      MACRO      ;DECREMENT A  C,V destroyed
        SEC
        SBC          #1      ;A:=A-1
        CLC
        ENDM

PHX      MACRO      ;PUSH X ON STACK
        FCB          $2C
        NOP          ;Skip 2 bytes
        RES          1,0
        STA          $-1
        TXA
        PHA
        PHP
        LDA          $-7
        PLP
        ENDM

PHY      MACRO      ;PUSH Y ON STACK
        FCB          $2C
        NOP
        _ASAVEY RES 1
        STA          _ASAVEY
        TYA
        PHA
        PHP
        LDA          _ASAVEY
        PLP
        ENDM

PLX      MACRO      ;GET X FROM STACK
        FCB          $2C
        NOP
        _XASAVE RES 1
        STA          _XASAVE
        PLA
        TAX
        PHP
        LDA          _XASAVE
        PLP
        ENDM

PLY      MACRO      ;GET Y FROM STACK
        FCB          $2C
        NOP

```

```

_YASAVE RES 1
STA _YASAVE
PLA
TAY
PHP
LDA _YASAVE
PLP
ENDM

endif

if      prosel==zero      ;set cmos option
opt     C02
endif

opt     lis
END

```

VII. Macro's voor andere telefoonnummers dan de eerste pagina van vitel.dir

Het is inderdaad mogelijk om aan ieder telefoonnummer een macroblok te koppelen. Als men alleen macro's gebruikt voor de eerste twintig telefoonnummers uit de telefoonboekfile Vitel.dir gaat alles automatisch. Heeft men hier niet genoeg aan, dan kan men bv voor pagina twee (21..40) als volgt te werk gaan:

```

Copy vitel.dir naar vitel.tijdelijk,
Verwijder de eerste 20 nummers uit vitel.dir met ed.
copy vimacro.def naar vimacro.tijd,
Run defmac.
Rename Vimacro.def naar Vimacro2.def,
Delete Vitel.dir en rename vitel.tijdelijk weer naar Vitel.dir,
Men kan nu nadat pagina 2 van vitel.dir geprint is op het scherm, een breek
geven en met !load vimacro2.def 340,7ff de nieuwe macrofile laden. Dit is
echter niet voldoende, men moet ook nog de variabele PAGE $01 maken, (zie
file Vidi.mac), en na afloop weer terugzetten op de oude waarde.

```

VIII. Tijd printen op de statusregel

Daar er geen standaard entry in de jumptabel van io/dos-65 zit om de tijd op de statusregel te printen, heb ik zelf de routines daarvoor in viditel opgenomen. Op 1 MHz systemen is duidelijk een hinderlijk knippen van het beeldscherm te zien als de cursor op de statusregel gepositioneerd wordt, de tijd daar geprint word, en vervolgens de cursor weer teruggezet wordt op de oorspronkelijke plaats, ergens tussen regel 1 en 24. Om dit zo onopvallend mogelijk te doen kijkt de routine UPTIM allereerst of de correcte tijd al geprint is door vergelijking van SECONDS met CLSEC, is dat niet het geval, dan wordt nog eens vergeleken of CLSEC5 de waarde vijf bereikt heeft. Indien dat zo is, wordt de tijd pas geprint. De macro MOVEQ 5,CLSEC5 zorgt ervoor dat de tijd iedere vijf seconden op het scherm geprint wordt. Wil men de tijd iedere seconde bijgewerkt zien moet er MOVEQ 1,CLSEC5 staan, of zoals in onderstaande listing MOVEQ 3,CLSEC5 om de tijd iedere drie seconden te printen.

```

;Time is only printed every 5 sec.
UPTIM LDA SECONDS
CMP CLSEC ;Print time if not equal
BEQ 33.F
STA CLSEC ;Save current value
DEC CLSEC5
BNE 33.F

```

```

MOVEQ    3,CLSEC5           ;***Tijd iedere 3 seconden printen***
JSR      CUROFF
JSR      CURSAVE
JSR      INVVID             ;Set inverse video
JSR      PRISEC             ;Seconds
BNE      32.F               ;Update only if 00
JSR      PRIMIN             ;Minutes
BNE      32.F               ;Update only if 00
PRIHOUR  LDA      HOURS      ;Hours
LDX      #7
JSR      POSTAT
JSR      TIOU1
32      JSR      NORVID      ;Normal video
JSR      CURLOAD
JSR      CURON
33      RTS
<end>

```

Nagekomen aanvulling.

IX. Editen met viditelpagina's =====

Met de nieuwe viditel is het mogelijk om een enkele pagina afzonderlijk op disk te zetten. Met append kan men dan zelf een file samenstellen met de pagina's die men bij elkaar wil groeperen. Al met al is dit nogal een omslachtige manier. Indien men alleen pagina's wil VERWIJDEREN kan dit ook direct vanuit de viditel terminalmode met de volgende truc:

Laad de file waaruit een of meer pagina's verwijderd moeten worden met L File. Ga met C naar de terminalmode. Iedere pagina die men wil bewaren kan men opslaan door -als hij in het pagebuffer staat- de volgende bewerking uit te voeren: ^K^L ^W ^Z ^S. De geload data word dan overschreven door de nieuwe data (check met F-key) en kan met S File op disk gezet worden.

<end>

De Eprommer: levering helaas vertraagd.

In het Juninummer van de 6502 Kenner werd de Eprommer gepubliceerd. Verder is er toen verteld dat de print en de PAL via Nico de Vries besteld konden worden. Op dat moment gingen we er van uit dat de print geleverd zou kunnen worden op het moment dat de Kenner in de bus zou liggen. Helaas zijn er bij de productie van de prints enkele vertragingen opgetreden. De situatie is nu zo dat we verwachten binnenkort (binnen enkele weken) met de levering te kunnen beginnen. Mocht dit niet het geval zijn, dan zal het reeds betaalde geld teruggestort worden.

Adri Hankel/Nico de Vries.

DE 6502 KENNER

DOS-65 Corner

```

; file : IRQREST.MAC
; auteur : adri hankel
; datum : 21-sep-88
; funktie : herstellen systeemcontrole na 'crash'

; Deze routine maakt gebruik van de bij DOS65 niet gebruikte NMI vector.
; Bij het ontwikkelen van een programma kan het voorkomen, dat het
; systeem 'hangt'. Dit is onder meer het geval bij een ongecontroleerde
; loop, of een programma waar de interrupt flag gezet wordt. Op dat
; moment wordt de keyboard interrupt niet meer verwerkt, ^C werkt dus
; niet meer.
; Door deze routine resident in het geheugen te zetten, en een NMI te
; genereren, kan men in de meeste gevallen weer de controle over het
; systeem verkrijgen.
; Voorwaarden voor een goede werking zijn:
; - de NMI lijn moet aangesloten zijn, zoals beschreven in de DOS65
; hardware manual (pagina 3, en fig. 4, pagina 5).
; - De DOS dient aanwezig te zijn, omdat deze routine eindigt met
; een jump naar de 'warme start' van DOS65.

; Gebruik van deze routine:
; - ASsembleer IRQREST.MAC
; - laad IRQREST.BIN (OLO IRQREST.BIN 0)
; Het verdient aanbeveling IRQREST.BIN automatisch te laden, door
; OLOAD IRQREST.BIN 0 op te nemen in LOGIN.COM

; *** externe adressen ***
E133 aciactmd equ $e133 ;acia commando register
E7B1 nmivec equ $e7b1 ;NMI vector
E7B5 irqvec equ $e7b5 ;IRQ vector
C003 dos equ $c003 ;warme start DOS65
F22F irqdest equ $f22f ;start adres IRQ routine IO65
E10E vaier equ $e10e ;via a int. enable register
E11E vbier equ $e11e ;via b int. enable register

; *** zet NMI vector op begin routine ***
E7B1 A0E5 org nmivec
fdb irqrest

E5A0 org $e5a0 ;resident geheugen

E5A0 D8 irqrest cld ;reset decimal mode
E5A1 78 sei ;geen interrupts
E5A2 A9 2F lda #irqdest&255 ;herstel interrupt vector
E5A4 8D B5E7 sta irqvec
E5A7 A9 F2 lda #irqdest>>8
E5A9 8D B6E7 sta irqvec+1
E5AC A9 C2 lda #$c2 ;systeem via's interrupt enable
E5AE 8D 0EE1 sta vaier
E5B1 A9 80 lda #$80
E5B3 8D 1EE1 sta vbier
E5B6 A9 00 lda #$00 ;acia cmd op default waarde
E5B8 8D 33E1 sta aciactmd
E5BB A2 FF ldx #$ff ;stack reset
E5BD 9A txs
E5BE 58 cli ;interrupts weer ok
E5BF 4C 03C0 jmp dos ;spring naar DOS

E5A0 end irqrest

```

Door: Nico de Vries

Het is alweer geruime tijd geleden, dat bovengetekende een spreekbeurt heeft gehouden over programmeerbare logica. Inmiddels heeft de tijd niet stilgestaan: er is tegenwoordig een groot aantal soorten programmeerbare logica op de markt, terwijl de gebruiksgraad inmiddels zodanig is dat rustig van standaardcomponenten gesproken mag worden. Voeg hierbij het feit dat deze soort componenten ook hun intrede in de club hebben gedaan (in de EPROM-programmer zit een PAL) en u heeft alle redenen bij elkaar waarom dit verhaal is gemaakt.

1. Inleiding.

Hoe zat het ook alweer? In de logica kent men slechts twee toestanden, die meestal met 'waar' en 'niet-waar', 'true' en 'false', '1' of '0' of in analogie met de meest gebruikelijke spanningsniveaus 'hoog' en 'laag' worden aangeduid. De gebruikte vorm van logica is TTL, dat Transistor-Transistor-Logica betekent. Deze familie werkt met een voedingsspanning van 5 Volt. Nu ligt het voor de hand om dat af te spreken dat 5 Volt een '1' of 'waar' is, en nul Volt een '0' of 'niet-waar'. Dit is niet helemaal juist. In de praktijk is een '0' een spanning tussen 0 en 0.8 Volt en een '1' een spanning tussen 2.4 en 5 Volt. Deze grenzen worden namelijk bepaald door de karakteristieken van de in- en uitgangstrappen.

Een heleboel dingen uit het dagelijks leven kunnen worden gevangen in deze '0' en '1' terminologie. Het regent bijvoorbeeld, met als tegengestelde het regent niet. Het wordt echter pas logica als er verbanden tussen diverse logische verschijnselen worden gelegd. Bijvoorbeeld: Ik draag een jas als het koud is, OF als het regent. Een ander voorbeeld: Het tocht in huis als het raam EN de deur openstaan. De met hoofdletters afgedrukt worden geven het verband aan tussen twee logische grootheden: koude OF regen, raam EN deur. Het resultaat is weer een nieuwe logische grootheid: Het al dan niet dragen van een jas, of het wel of niet tochten in huis.

De twee genoemde verbanden EN en OF (of in het Engels: AND en OR) vormen de basis van de zogenoemde Booleaanse logica. De Engelsman Boole heeft dit soort verschijnselen namelijk uitgebreid bestudeerd en in een wiskundig model gegoten: de Booleaanse algebra. Deze algebra maakt onder andere gebruik van waarheidstabellen die er voor de voorbeelden als volgt uitzien:

JAS	KOUD	REGEN	TOCHT	RAAM	DEUR
0	0	0	0	0	0
1	1	0	0	1	0
1	0	1	0	0	1
1	1	1	1	1	1

Deze twee verbanden, AND en OR vormen de basis van de Booleaanse algebra. Boole heeft namelijk aangetoond, dat ieder willekeurig verband tussen een willekeu-

rig aantal logische grootheden altijd kan worden uitgedrukt in een combinatie van deze twee bewerkingen. Merk overigens op, dat een AND-verband tussen twee variabelen hetzelfde is als een binaire (tweetallige) vermenigvuldiging, en dat een OR bijna overeenkomt met een binaire optelling. Een AND-verband wordt ook wel een product, en een OR-verband een som genoemd. In de Booleaanse algebra schrijft men voor een AND dan ook meestal een punt of een sterretje, en voor een OR een plus. De voorbeelden worden dus zo genoteerd:

JAS = KOUD + REGEN
TOCHT = RAAM * DEUR

Eerst een opmerking over de OF-bewerking. Het voorbeeld klopt: men draagt een jas als het koud is, of als het regent, maar ook als het koud is en het regent. Dit is wat tegenstrijdig met het dagelijks taalgebruik van het woord of, waarmee men meestal een keuze van 1 uit 2 bedoelt. Deze bewerking bestaat echter ook in de Booleaanse algebra en heet de exclusieve OF, of in verkort Engels: de EXOR. De schrijfwijze van de EXOR is niet genormaliseerd. De EXOR is een zuivere binaire optelling:

EXOR	A	B
0	0	0
1	1	0
1	0	1
0	1	1

We hebben nog 1 groundbewerking tegoe: de inversie of de NIET-bewerking. Deze bewerking vindt altijd plaats op 1 enkele grootheid, terwijl de andere drie steeds betrekking hebben op twee grootheden. De NIET-bewerking (of in het Engels: NOT) verandert eenvoudig 'waar' in 'niet-waar' en andersom. De schrijfwijze op papier is met een streep boven de naam:

REGEN dit is dus: niet-REGEN

Dit is voor computertoepassingen wat minder handig, vandaar dat men in dergelijke omgevingen meestal een slash gebruikt om inversies aan te geven: /REGEN. Ook bewerkingen kunnen worden omgekeerd: een AND wordt dan een NAND en

een OR wordt een NOR. De waarheidstabellen hiervan zijn:

NOR	A	B	NAND	A	B
1	0	0	1	0	0
0	1	0	1	1	0
0	0	1	1	0	1
0	1	1	0	1	1

2. DeMorgan.

De heer Boole was niet de enige die zich bezig hield met dit soort zaken. De Schot DeMorgan bijvoorbeeld ontdekte een belangrijke regel, die gebruikt kan worden bij het omzetten van OR-verbanden in EN-verbanden en andersom. Vergelijk de volgende twee vergelijkingen eens:

$$\text{JAS} = \text{KOUD} + \text{REGEN}$$

$$/\text{JAS} = / \text{KOUD} * / \text{REGEN}$$

Zoals te zien is, zijn beide regels in feite hetzelfde. We dragen geen jas, als het niet regent, en als het niet koud is. Vergelijk ook eens:

$$\text{JAS} = / \text{WARM} + / \text{DROOG}$$

$$/\text{JAS} = \text{WARM} * \text{DROOG}$$

De regel van DeMorgan is dan ook: Als de uitgangsvariabele geïnverteerd wordt, worden alle ingangsvariabelen eveneens geïnverteerd, en verandert een AND-bewerking in een OR en een OR-bewerking in een AND. Dit wordt iets moeilijker als er combinaties van ANDs en ORs in de vergelijking voorkomen, maar hiervoor zijn een aantal rekenregels beschikbaar die de volgorde van werken bepalen, iets in de trant van 'Mijnheer Van Dalen Wacht Op Antwoord'. Het valt echter buiten het bestek van dit verhaal om al deze regels uit te leggen.

De belangrijkste conclusie die uit de stelling van DeMorgan getrokken kan worden is dan ook: ieder willekeurig logisch verband tussen een aantal logische variabelen kan worden genoteerd als een som van producten, of als product van sommen. Een voorbeeld:

$$U = A * /B \text{ is gelijk aan: } /U = (/A + B) * /C$$

(De rekenregels eisen ook in Booleaanse algebra dat een product wordt uitgewerkt voor een som, vandaar de haakjes in de rechter vergelijking. Schrijft men de rechter vergelijking uit, dan staat er:

$$/U = (/A * /C) + (B * /C).$$

3. Programmeerbare Logica.

Wat heeft dit alles nu te maken met programmeerbare logica? Alles. Het voorgaande was nodig om te laten zien waarom pro-

grammeerbare logica is, zoals zij is. Maar nu de praktijk.

Een ieder die wel eens wat uitgebreider heeft gespeeld met TTL-poorten heeft waarschijnlijk wel ontdekt, dat de wat complexere logische schakelingen relatief veel TTL-ICs vereisen: in 1 IC met 14 of 16 pootjes zitten 4, 3, 2 of zelfs maar 1 poort(en). Bij wat grotere aantallen ingangen en/of complexiteiten zit je al gauw met een kwart Eurokaart aan oppervlak. Nu was dat 15 jaar geleden nog niet zo erg: TTL was zo ongeveer het meest geavanceerde wat men krijgen kon. In de hedendaagse microprocessorwereld echter, lijkt zo'n kwart Eurokaart al gauw verspild: een 40 pins CPU is aanmerkelijk krachtiger dan een clustertje TTL. Ook de wens om steeds meer functies op minder ruimte onder te brengen werd door de lage integratiegraad van TTL in de grond geboord. Toch kunnen we nauwelijks zonder die TTL: er zal toch gede-codeerd, gebufferd en geconverteerd moeten worden.

Deze gedachten, gecombineerd met de wetenschap dat alle logische verbanden als som van producten kunnen worden beschreven, leidde in 1979 tot de uitvinding van de PAL, een afkorting van Programmable Array Logic. Degene die de uitvinding deed heette John Birkner en hij werkte toen bij MMI (Monolithic Memories Inc.). De opbouw van zo'n PAL is eigenlijk heel eenvoudig. In afbeelding 1 is de eenvoudigste PAL afgebeeld: de PAL10H8.

Deze PAL is uitgerust met 10 ingangen, die op de pinnen 1 t/m 9 en pin 11 te vinden zijn. De 8 uitgangen zitten op de pinnen 12 t/m 19. De pinnen 10 en 20 zijn respectievelijk de massa (GND) en de voedingspanning (Vcc). Deze PAL heeft dus 20 pinnen en zit in dezelfde behuizing als bijvoorbeeld een 74LS244. Het aantal in- en uitgangen zit verstopt in het typenummer: 10H8 geeft aan 10 ingangen en 8 uitgangen, de H vertelt ons dat de uitgangen actief hoog zijn.

Deze PAL (alle PALs trouwens) is gebaseerd op het som van producten principe. De som wordt gevormd door en OR-poorten aan de uitgangen, die bij deze PAL ieder 2 ingangen hebben. Deze PAL kan dus per uitgang een logisch verband genereren dat bestaat uit de som van twee producten. De ingangen van de OR-gate zijn op hun beurt verbonden met de uitgangen van twee AND-poorten. Deze AND-poorten zijn wat afwijkend getekend, maar ze hebben ieder 20 ingangen. Om niet alle ingangen te hoeven tekenen, worden ze voorgesteld door de liggende lijnen in de tekening. 1 liggende lijn is dus eigenlijk een bundel van 20 verbindingen waartussen een AND-verband kan bestaan. Daarom wordt zo'n lijn ook wel een productterm genoemd. Iedere ingang van de AND-poort is verbonden een pull-up weerstand naar de plus en met 1 van de staande lijnen in de tekening. Deze verbinding is echter niet vast, maar bestaat uit een heel

klein stukje metaal op de chip, dat dezelfde eigenschappen heeft als een zekering: als je er teveel stroom doorstuurt, dan smelt het weg. Een dergelijke verbinding wordt in het Engels een 'fusible link' of kortweg 'fuse' of 'link' genoemd.

De staande lijnen in de tekening zijn verbonden met de ingangen. De ingangsbuffers van iedere pin hebben twee uitgangen: een normale en een inverterende. Omdat er tien ingangen zijn, zijn er dus twintig staande lijnen. Op de kruising van iedere staande lijn met een liggende lijn is dus steeds 1 fusible link aanwezig. Als men de PAL koopt, zijn alle links aanwezig. In een 10H8 zijn dit er 20 (ingangen \times 2) maal 8 (uitgangen) maal 2 (aantal producttermen per uitgang) is 320 stuks. Het gebied op de tekening waar zich de fusible links bevinden (de kruisingen tussen de lijnen) wordt het fuse-array of kortweg het array genoemd. Door nu bepaalde links weg te programmeren met een grote stroom kunnen bepaalde verbindingen verbroken worden, en kan een logisch verband tussen de in- en uitgangen worden gecreeerd.

4. Hoe werkt een PAL nu?

Een voorbeeld. Stel we willen de volgende logische functie in de PAL programmeren:

$$U = A * B + C$$

Stel, we noemen pin 19 U, pin 1 A, pin 2 B en pin 3 C. Eerst moet er een verbinding komen tussen de inverse van pin 1 en de bovenste AND-poort. Dit is de kruising tussen de staande lijn 3 en de liggende lijn 0. De tweede verbinding is die van pin 2 naar eveneens de bovenste AND-poort. Dit is de verbinding tussen de staande lijn 0 en de liggende lijn nul. Programmeert men nu alle andere verbindingen tussen de staande lijnen en de liggende lijn 0 weg, dan heeft de liggende lijn 0 de functie $A * B$ gekregen. De tweede productterm gaat precies zo. De OR-poort aan de uitgang zorgt voor het OR-verband, zodat de functie C op de tweede liggende lijn moet komen. Dit is de kruising tussen de staande lijn 4 en de liggende lijn 1. Programmeert men op deze liggende lijn alle andere verbindingen weg, dan is de gewenste functie gemaakt.

Wat gebeurt er nu, als we niet alle producttermen in de PAL gebruiken? Dan blijven alle verbindingen op die liggende lijn intact. Er ontstaat dan in een PAL10H8 de volgende vergelijking op die productterm:

$$1 * /1 * 2 * /2 * 3 * /3 * \dots$$

Iets anders geschreven:

$$(1 * /1) * (2 * /2) * (3 * /3) * \dots$$

Als we nu in de waarheidstabel voor een AND kijken, dan zien we dat $(1 * /1)$ altijd nul is. Kijken we nog verder, dan blijkt dat $0 * \text{IETS}$ altijd nul oplevert. Zo'n productterm met alle links intact is dus altijd logisch nul. Kijken we vervolgens naar de waarheidstabel voor een OR, dan vinden we dat $0 + \text{IETS}$ altijd IETS oplevert. Het netto resultaat is dus, dat een productterm met alle links intact geen invloed op de uitgang heeft.

5. Nog meer PALs.

De PAL10H8 is natuurlijk niet de enige PAL die er te koop is. Zo kan het aantal uitgangen en het aantal ingangen veranderd worden, waarbij de som van die twee steeds 18 blijft (vanwege de 20 pins behuizing). Zo zijn er dus ook een 12H6, een 14H4 en een 16H2. Ook kun je besluiten de PAL niet te voorzien OR-poorten in de uitgangen, maar van NOR-poorten. De H in het typenummer wordt dan een L (van actief laag). Op die manier ontstaan de 10L8, 12L6, 14L4 en 16L2 PALs.

In de praktijk bleek al spoedig, dat de behoefte aan actief lage PALs veel groter was dan aan actief hoge. Ook was het aantal producttermen vaak te klein. Verder was er behoefte aan PALs die aan een bus gehangen konden worden, of anders gezegd, de uitgangen moesten ook in tri-state kunnen worden geplaatst. Dit leidde tot een tweede reeks PALs, waarvan de PAL16L8 en de PAL16R4 zijn afgebeeld.

In de PAL16L8 zijn alle uitgangen actief laag (een PAL16H8 bestaat niet van MMI). Verder valt op, dat alle uitgangen voorzien zijn van 8 producttermen. De bovenste productterm echter wordt niet gebruikt voor de logische functie van uitgang, maar voor de tri-state enable ervan. Een laatste nieuwigheidje is, dat de 6 middelste uitgangen teruggekoppeld zijn. Deze 6 terugkoppelingen zijn er verantwoordelijk voor, dat de naam 16L8 ontstaat: het array heeft immers 16 ingangen. Een telling leert, dat een PAL16L8 in totaal 2048 fuses bevat, aanmerkelijk meer dan de PAL10H8 van daarstraks. De PAL16L8 is de werkezel onder PALs: deze PAL wordt meer gebruikt dan alle andere PALs bij elkaar!

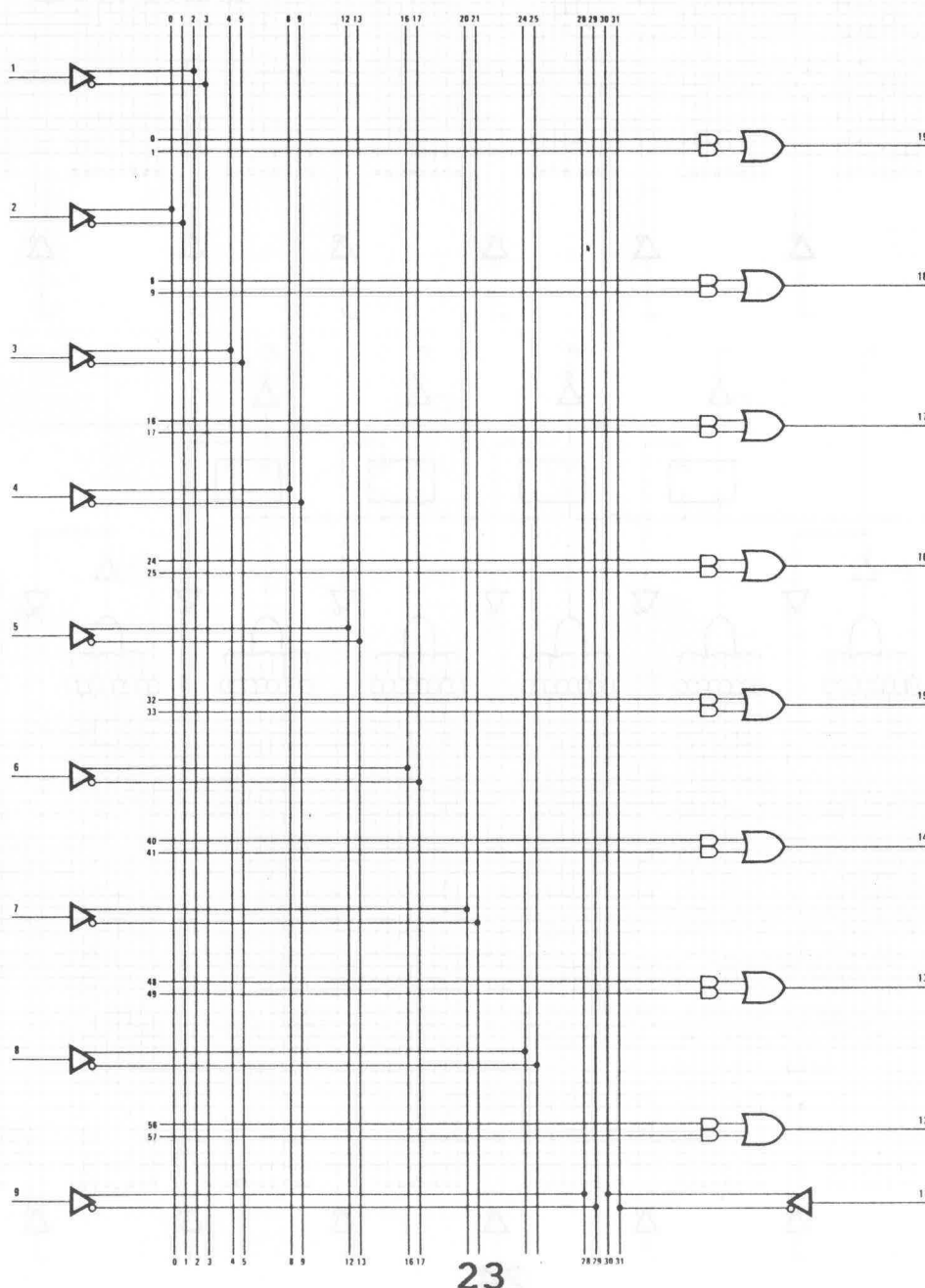
De PAL16R4 is een broertje van de PAL16L8: het array is precies even groot. Deze PAL bevat echter registers in 4 van de uitgangen. Omdat deze registers ook geclockt moeten worden, is hiervoor pin 1 gebruikt. De vrijgekomen array-ingang wordt nu de terugkoppeling van pin 19. Als bijzonderheid is verder te vermelden, dat pin 11 een tri-state enable voor de registeruitgangen is geworden: de vrijgekomen array-ingang is de terugkoppeling van pin 12 geworden terwijl de registeruitgangen nu 8 producttermen hebben. Er bestaan nog twee

varianten: de PAL16R6 en de PAL16R8 met respectievelijk 6 en 8 registeruitgangen.

We hebben reeds gezien dat er een 10L8 en een 10H8 bestaat. De volgende stap is geweest, om een PAL16L8 te maken, waarvan de uitgangspolariteit per uitgang bepaald kon worden: de PAL16P8. 1 uitgang van de PAL16P8 is in afbeelding 5 getoond. Tussen de OR-poort en de uitgangsbuffer is een EXOR-poort geplaatst met twee ingangen. De tweede ingang van deze poort is met de GND verbonden via een fuse. Als de fuse aanwezig is, is de PAL actief laag, wordt zij weggeprogrammeerd dan is de uitgang actief hoog (ga dit na in de waarheidstabel van de EXOR). Analoo

10H8

fig. 1



hieraan bestaan er ook 16RP4, 16RP6 en 16RP8 PALs.

Een zelfde verhaal kan worden gehouden voor PALs met 24 pinnen, ofschoon niet alle besproken varianten inderdaad bestaan. Een bijzonderheid van 24-pins PALs is, dat de behuizing niet 0.6 inch maar 0.3 inch breed is. Een aantal typenummers zijn: 12L10, 14L8, 20L8, 20R6, enzovoort.

In deel 2.....

De volgende keer worden de laatste ontwikkelingen op dit gebied belicht en wordt er ook aandacht geschonken aan andere soorten programmeerbare logica, alsmede aan de manier waarop de inhoud van de PAL wordt ontwikkeld.

Fig. 2

16L8

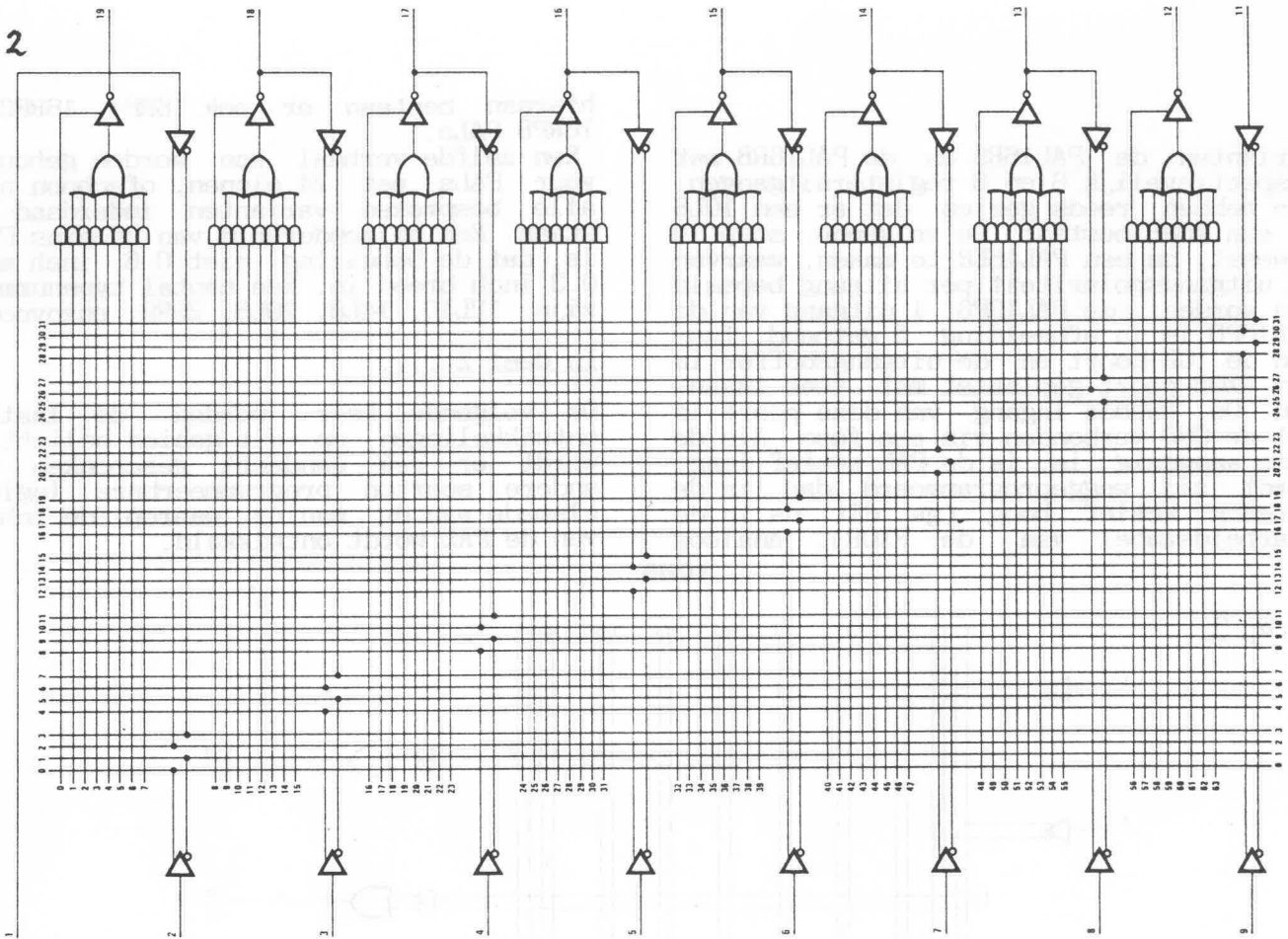


Fig. 3

16R4

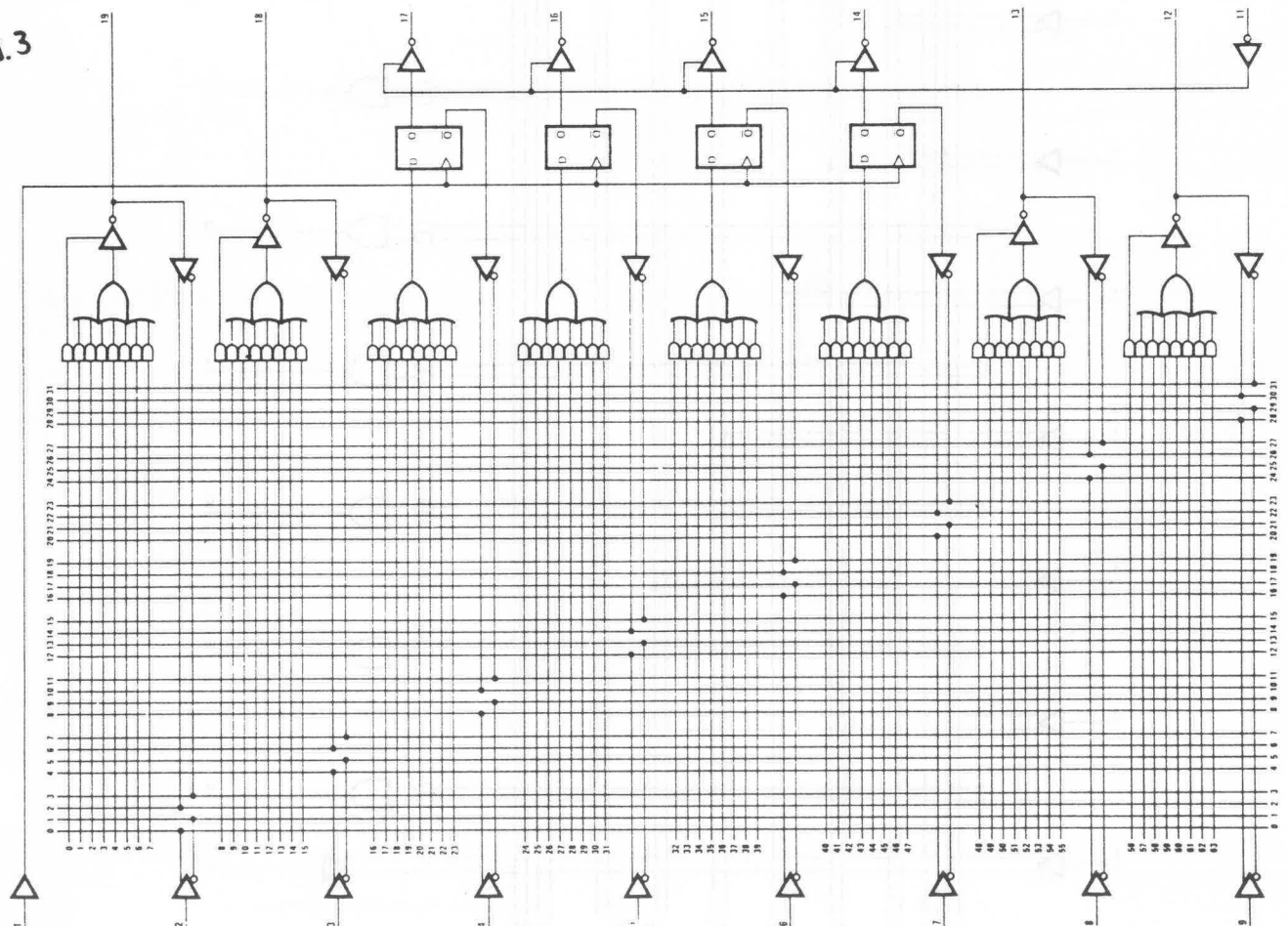
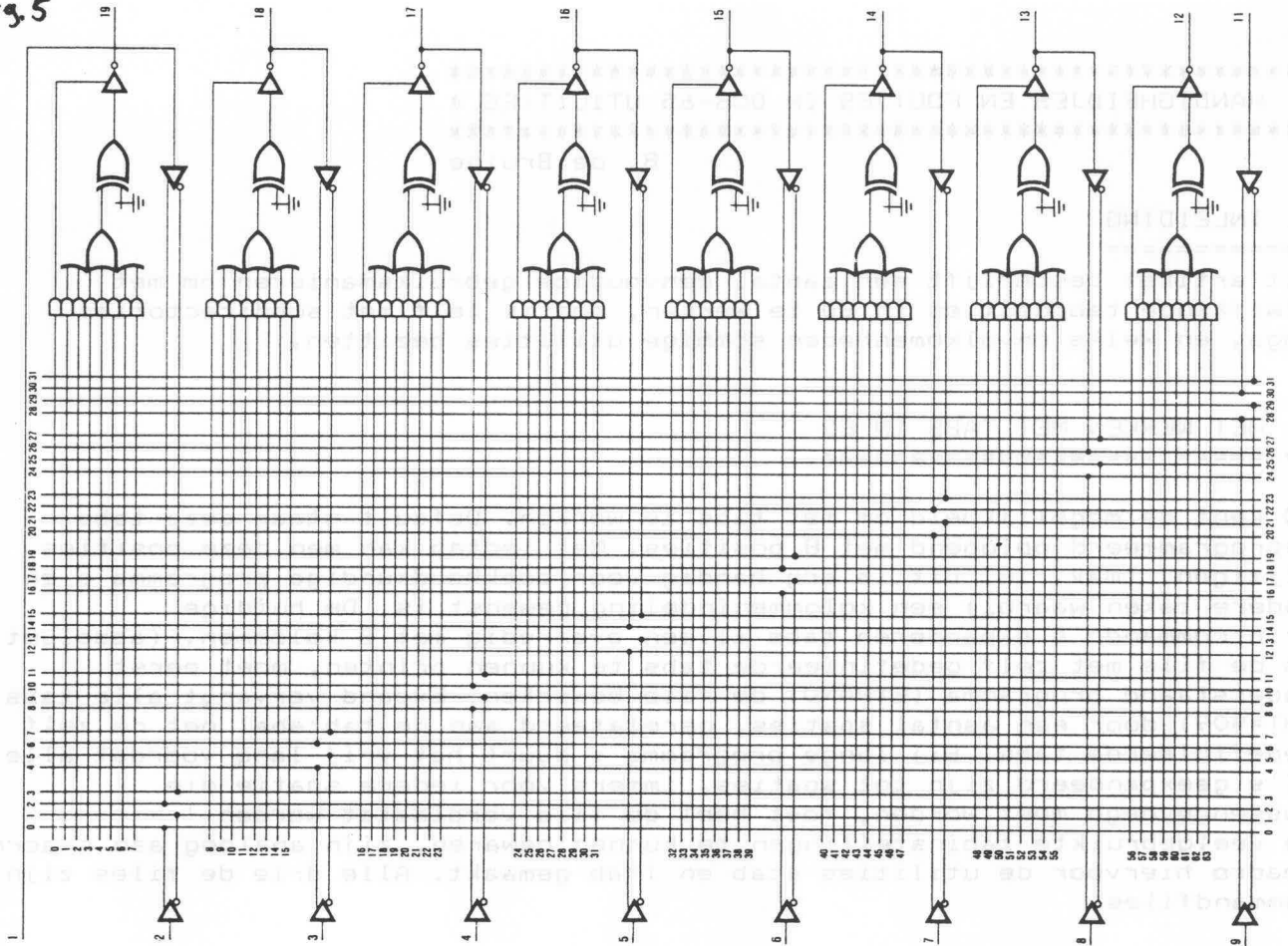
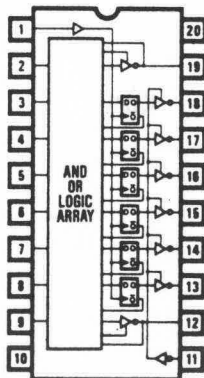


Fig.5

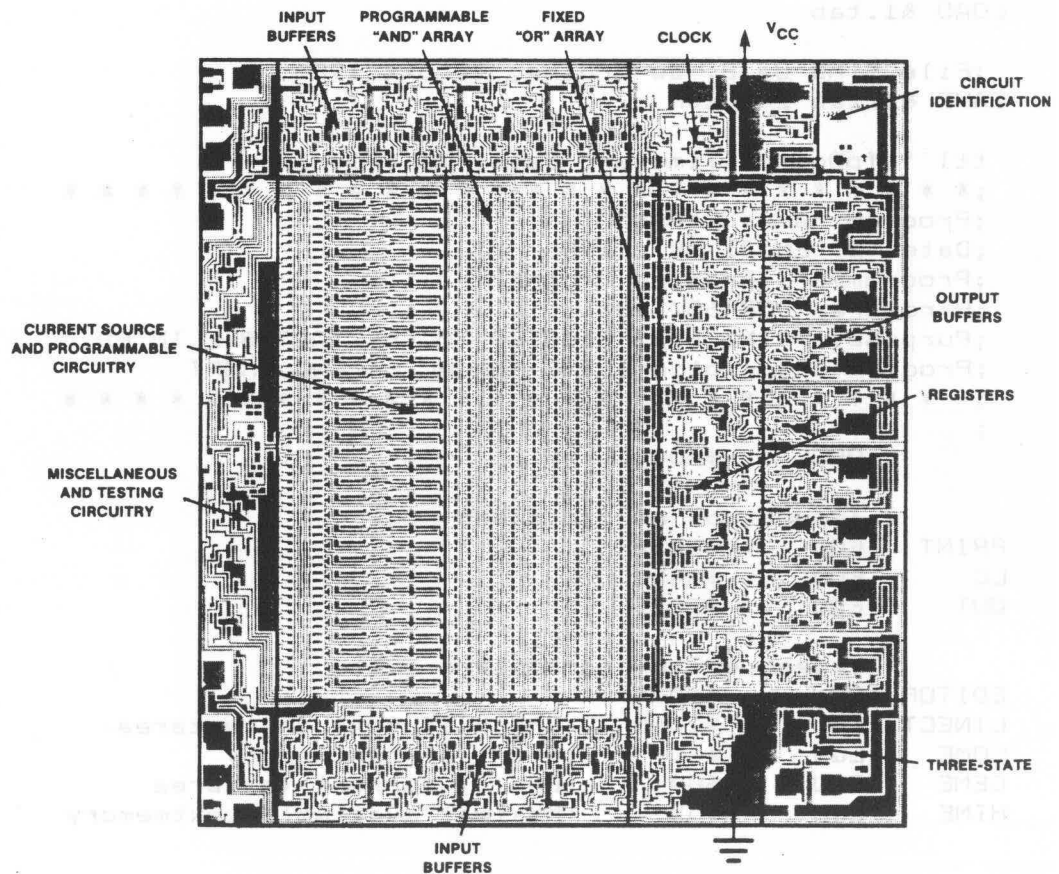
16P8



PAL16R6



PAL16R6 Logic Symbols



PAL 16R6 Metalization

```
*****
*  HANDIGHEIDJES EN FOUTJES IN DOS-65 UTILITIES *
*****
B. de Bruine
```

1. INLEIDING

=====

Dit artikel beschrijft een aantal eenvoudige gebruiksmanieren om met afwijkende tabposities in ED te werken, hoe ik zelf met subdirectories omga, en welke onvolkomenheden sommige utilities bezitten.

2. HET WERKEN MET TABS IN ED

=====

ED kent de mogelijkheid om met tabs te werken. Default staan deze tabs geprogrammeerd oplopend met 8 posities. Met !edtab kan men deze posities wijzigen. (max. 16) Dit is erg handig voor boekhoudkundige programma's en andere zaken waarbij een kolommenindeling gewenst is. De huidige printcommando's expanderen tabs alleen evenredig met 8 kolommen. (absoluut) Om de file met zelf gedefinieerde tabs te kunnen printen, moet eerst onderstaand programma (EXPAND) de file bewerken. Expand vervangt alle tabs (HT=\$09) door een aantal spaties, gerelateerd aan de tabtabel met de zelf gedefinieerde tabs. Bij lange programma's duurt het vrij lang voordat alle HT's geëxpandeerd zijn tot spaties, immers voor iedere spatie die tussengevoegd moet worden, moet heel de file verplaatst worden ! Om veelgebruikte tabinstellingen te kunnen bewaren, zijn analoog aan smacro en lmacro hiervoor de utilities stab en ltab gemaakt. Alle drie de files zijn commandfiles.

```
;File LTAB by B. de Bruine, 20 okt 1986
LOAD &1.tab
```

```
;File STAB by B. de Bruine, 20 okt 1986
SAVE &1.tab 200,20F
```

```
ttl "tfpDos 65 Time and date: %s file %s"
;*****
;Program name: EXPAND
;Date: 20 EN 21 OKT. 1986
;Programmer: Bram de Bruine
;Hardware: DOS 65/2.01
;Purpose: PRINT TABS RELATED TO THE TABTABLE IN ED.
;Program description: SEE MANUAL EXPAND.SHORT
;*****
;
```

Page %d"

```
ORG $A000

;dddd dos 65 i/o dddd

PRINT EQU $C03B
LC EQU $AA5C
OUT EQU $C023

;eeeeee editor parameters eeeee

EDITOR EQU $1000
LINECT EQU $1148 ;Check if text in textarea
LOME EQU $00 ;Start of textmemory
CEME EQU $0A ;End of current textarea
HIME EQU $05 ;Maximal available textmemory
```

```

TBBF EQU $200 ;Actual tabpositions
STACK EQU $100 ;User&system stack

;cccc constants cccc
HT EQU $09
ESC EQU $1B
ENDCOL EQU 81 ;Maximal right tabposition
NUMTAB EQU 16 ;Maximal allowed tabs

EXPAND LDY #0
      JSR INIT
      JMP START

;Vvvvv variables used by expand vvvv
TXT EQU $90 ;Points to actual fetched character
MEND EQU TXT+2 ;Actual endaddress of the textarea
SEND EQU TXT+4 ;Savebytes for mend
COL RES 1 ;xposition of current character in line
TAB RES 1 ;Indicates which tab is the actual tab (0..16)
SPINDEX RES 1 ;Number of spaces to be expanded
YP RES 1 ;Message pointer

SCANLIN JSR INCTXT
START INC COL
SCOMP LDA [TXT],Y
      BEQ EXIT ;End of file ?
      CMP #$0D
      BEQ NEWLINE ;New line ?
      CMP #HT
      BNE SCANLIN ;Horizontal tab ?
      JMP EXPHT

NEWLINE JSR RESVAR
      JMP SCANLIN

EXIT LDA TXT
      CMP LOME ;Is there a text present ?
      BNE QUIT
      LDA TXT+1
      CMP LOME+1
      BNE QUIT
      JSR PRINT
      FCC 'EMPTY TEXTBUFFER\`a\`', $0D, 0
      JMP QERR ;QF

QUIT JSR PRINT
      FCC $0D, 'FILE EXPANDED ..READY', $0D, 0
      JSR LINECT ;Update memory pointers /ed
      LDA #EDITOR&255
      STA LC
      LDA #EDITOR>>8 ;Switch LC to editor
      STA LC+1
      RTS

EXPHT INC TAB
      LDX TAB
      CPX #NUMTAB ;Tab overflow ?
      BMI TABFLOW
      JSR PRINT
      FCC 'UNDEFINED TAB\`a\`r\`', 0
      RTS ;Quit fast

TABPOS LDA TBBF, X ;Load tabposition
      CMP #ENDCOL ;Right tab border

```



```

BPL TABFLOW
CMP COL ;Tabpos <= col ?
BEQ EXPHT
BMI EXPHT
SEC
SBC COL ;Spindex=tabpos-col
STA SPINDEX ;Number of spaces
BEQ REPLAC
;If spindex=0 then replace ht by space
DEC SPINDEX ;Exclusive of ht
JSR SENDMEND
LDY #0
LDA [MEND],Y
LDY SPINDEX
STA [MEND],Y
JSR DECMEND
LDA TXT+1 ;Everything moved ?
CMP MEND+1
BNE MOVE
LDA TXT
CMP MEND
BNE MOVE
LDX SPINDEX
INX
LDY #0 ;Insert spaces
LDA #$20
STA [TXT],Y
JSR INCTXT
INC COL ;And update col
DEX
BNE INSERT
JSR UPDEND ;Update mend after expansion
JMP SCOMP ;Continue scanning
LDA #$20
STA [TXT],Y
JMP SCANLIN

;mmmmmmmmmmmm macrodefinition mmmmmmmmmmmmm
macro a,b ;copy b to a (a:=b)
LDA b ;a:=b
STA a
LDA b+1
STA a+1
RTS
endm

;***** subroutines *****
PHA
DEC MEND ;Mend:=mend-1
LDA MEND
CMP #$FF
BNE DECEX
DEC MEND+1
PLA
RTS

INC TXT ;Txt:=txt+1
BNE INCX
INC TXT+1
LDY YP

```

```

INY
STY YP
LDA BUSTEXT,Y
BNE SCREEN
LDY #$FF
STY YP
SCREEN JSR OUT ;Sign of busy
LDY #0
LDA TXT+1
CMP HIME ;End memory test
BNE INCEX
JSR PRINT
FCC 'OUT OF MEMORY ERROR',%OD,0
PLA ;Destroy returnaddr. of inctxt and qf
PLA
JMP QERR
INCEX RTS

BUSTEXT FCC 'PATIENCE IS A HONOURABLE THING ',0
TXTLOME COPY TXT,LOME ;Txt:=lome
MENDCEME COPY MEND,CEME ;Mend:=ceme
SENDMEND COPY SEND,MEND ;Send:=mend
UPDEND LDA SEND ;Update mend to current endaddress
CLC
ADC SPINDEX
STA MEND
LDA SEND+1
ADC #0
STA MEND+1
RTS

INIT JSR TXTLOME ;Setup pointers
JSR MENDCEME ; and initialise variables
LDY #$FF
STY YP ;Reset busy textpointer
JSR RESVAR
RTS

RESVAR LDY #$FF
STY TAB ;Reset actual tabcounter
INY
STY COL ;Reset character counter
RTS
END EXPAND

```

Expand vervangt alleen tabs door spaties die in het ramgeheugen van de editor aanwezig zijn. Wil men een file bewerken die niet in het werkgeheugen van ED past dan moet met ESC C Update en/of IN/OUT commando's steeds een nieuw stuk tekst ingelezen worden.

Expand wordt 'gerund' vanuit de commandmode van ED, daarna kan het op schijf gezet worden, en geprint worden op de printer met een normaal dos-65 commando.

3. SUBDIRECTORIES

=====

Om een subdirectory te bevorderen tot de defaultdirectory moet men gebruik maken van het ASN-commando. Om bijvoorbeeld van directory A de defaultdirectory te maken tikt men: ASN u=A/
Schakelt men veel om van subdirectory, dan is het handig om hiervoor een commandfile te maken:

A asn u=a/

B asn u=b/

enz. Alleen @, d en g geven problemen omdat het afkortingen zijn van bestaande commando's, gebruik deze dirs dan niet. Men kan nu door simpelweg 'B' in te tikken, directory B op de userdrive defaultdirectory maken.

4. SYSTEEM FOUTEN EN/OF ONGEMAKKEN

=====

laatste update: 28 jan 1988.

1. I/O-65

=====

- a. Cursorbesturing I/O 65 /Graphics Controlcodes voldoen allemaal aan de huidige gangbare standaard, behalve de cursor home code. Dit is \$1C i.p.v. \$1E. Wil men standaard zijn met andere machines, dan kan men het beste zowel op \$1C als \$1E een JSR HOME uitvoeren.
- b. Direct cursoradressering met ^T gaat bij Dos-65 nog steeds fout. (Remedie: Gebruik IO65 printcommando's 6502 kenner nr 53.)
- c. Zijn de Graphics ontleend aan een standaard ?? Het zou leuk zijn als ze bv Viditelcompatible zouden zijn. Dit is heel goed mogelijk. Alleen ESC, LF, FF en CR zouden hun oorspronkelijke code (kunnen) behouden. Inmiddels heb ik zelf een karaktergenerator gemaakt volgens de internationale videotexstandaard. Inloggen op viditeldatabanken is nu zonder meer mogelijk, en (grafische) basicode 3 ligt nu ook binnen het bereik van dos-65 !
- d. CLOCK Het zou erg handig zijn als er een variabele in IO/65 zou zijn die men kan testen op de systeem clockfreq. Bv 1=1mhz, 2=2mhz etc. Bij het maken van programma's kan dan dit byte gesampled worden, en kan men vertragingsslussen (bv) aanpassen aan de clockfrequentie.
- e. Ik mis een entry in de jumptabel om de tijd op de statusregel te printen.

2. Dos-65

=====

- a. DOS-65 wil dc. BV bij DIR *.mac* wordt geen match gevonden als de "c" van mac het veertiende karakter is !!! Er wordt dus 1 karakter te weinig gecheckt.
- b. Bij output redirect (> -tf) mis ik een optie om grafische attributen eruit te kunnen filteren. Voorbeeld: > -tf weg mon
MON> h 400,4ff

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0400 :	80	00	00	00	80	00	18	50	17	7F	80	C0	00	30	00	A0
0410 :	4C	20	C0	4C	23	C0	4C	12	CA	4C	1E	CA	4C	18	F0	4C

L L#LFrGLF~GLF~GL
Duidelijk is te zien dat schermattribuutcodes als ESC F, ESC G, ESC n, ESC i, enz alleen maar ontdaan worden van de ESC, terwijl de vervolgcode niet gefilterd word. Dit komt omdat MON de grafische mode aan/uit schakelt om de graphic (bv \$80 op locatie \$400) te kunnen printen. Mon genereert dan: ESC F ` ESC G. Om de hexdump met Ed te kunnen bewerken

zou alleen de ' geprint moeten worden. > -tfg weg mon, met -g: Filter schermcodes.

- c. Als de centronicsprinter 'not ready' is, hangt het systeem. Beter zou het zijn om een time-out beveiliging in te bouwen en/of een test of de breektoets ingedrukt is.

3. Utilities

=====

- a. Copy: Bij copy komt het bij mij regelmatig voor dat na afloop beide schijven dezelfde naam gekregen hebben. Dus copieer ik file 1 van schijf 1 naar schijf 2, dan hebben (soms) na afloop de schijven beiden de naam schijf 1.
- b. Help. Mag help geen info geven over utils die niet op schijf s: staan ?

4. ED:

=====

- a. Direct nadat de editor geladen is, komt er soms nogal wat rommel op het scherm. Na een ^R is alles weer ok.
Ook search (esc F,A) gaat dan niet goed. (data verkeerd op scherm ?) cursor wordt een paar regels voorbij het te zoeken woord geplaatst. Bij de volgende ESC : is alles weer in orde.
Dit is waarschijnlijk een scrollerror. (geconfig. voor 6845)
- b. Wordt in de commandmode van ED een utility gerund, bv !SEE file, dan heeft een ^C tijdens de uitvoering van SEE tot gevolg dat er teruggekeerd wordt naar Dos-65. De editfile is dan verloren ! Men moet in dergelijke situaties een ^Z geven. Dit is niet duidelijk gedocumenteerd.
- c. Treedt er een fout op (bv file not found/read error) na de aanroep ED FILE, dan gaat ED direct in de schermmode, zodat de foutmelding niet meer te lezen is. Oplossing: Check eerst met ESC Z of de hele file wel ingelezen is. Mooier zou het zijn als ED na een error in de commandmode blijft, zodat men de foutmelding kan lezen.
- d. Als er geen plaats meer is voor .b in de filenaam, maakt ed na een EX command een backup.edd file. Het zou handig zijn als Ed even een melding geeft in de trant van 'backupp.edd in stead of .b' zodat de gebruiker desgewenst zelf kan renamen. De volgende backup van een andere file met lange filenaam overschrijft namelijk de vorige.
Deze fout heeft misschien iets te maken met fout 2.a van dos-65 ?
- e. ESC U is erg handig bij lange files, tenminste als de goede schijf in de drive zit. Heeft men per ongeluk een andere schijf in de drive zitten, waar de filenaam van de te editen file niet opstaat, en men geeft een EX dan krijgt men de melding: output.edd not found. gaan we met 'C' terug om te kijken wat er nog in het geheugen staat, dan blijkt dit gewist te zijn!! Omdat ik de diskdrive hoorde schrijven, ben ik gaan zoeken naar de file. Deze bleek gewoon over een aantal andere files heengeschreven te zijn, ZONDER dat een filenaam was toegekend aan die file. Kortom: de hele disk is een puinhoop.
Waarschijnlijk gaat ED er vanuit dat de oorspronkelijke schijf nog in de drive zit, en schrijft hij -zonder enige controle- de rest van de file weg op de nieuwe schijf, exact op de plaats waar hij het zou schrijven als er geen schijven verwisseld zouden zijn. Dit werkt zeer demotiverend..

5. Mon 65

=====

- a. MON65: Vervang het E(xecute) commando door G(o). Verwarring tussen Dos en Mon commando's zijn dan uitgesloten! Vervang de ??? door FCB \$.

b. MON65 2.06: Let op dat een breakpoint na het runnen van een programma waar hij tegen dat breekpunt oploopt, automatisch verwijderd wordt. Dit is soms heel handig. In loops is het minder handig.

c. Idee: Display hexdump op scherm met vaste lay-out, dwz begin altijd met kolom 0: Dus H 403,ffff word dan als volgt gelist:

```

      0   1   2   3   4   5
400          xx  xx  xx      enz.
410
```

Dus op 401/2/3 staat gewoon een spatie. Dit leest wat makkelijker.

6. AS

=====

a. Pas op bij conversie van moserfiles naar AS, of bij toekenning van een waarde aan een label, waarbij die waarde pas verderop in de file bekend wordt. As maakt ze dan 00 en geeft geen foutmelding !! Moser slikt dit allemaal, maar AS niet !! (As kent geen forward label referencing)

;File testlabel

;Deze filetoont aan dat AS bij een toekenning van een waarde aan een label, ;deze waarde vooraf bekend moet zijn.

```

      A000          ORG      $A000

      E000  OUT      EQU      $E000

      0008  LABEL    EQU      TXT-MESSAGE ;****LABEL = 8 ****

A000 A2 00          LDX      #LABEL ;**** LABEL = 0 !! ****
A002 BD 0EA0  LOOP    LDA      MESSAGE,X
A005 20 00E0        JSR      OUT
A008 E8            INX
A009 C9 03          CMP      #$03          ;Etx ?
A00B D0 F5          BNE      LOOP
A00D 60            RTS

A00E 5741524E49      MESSAGE FCC      'WARNING',3
A013 4E4703
A016 4552524F52      TXT      FCC      'ERROR',3
A01B 03
                        END
```

global labels

```

LABEL      0000  LOOP      A002  MESSAGE      A00E  OUT      E000  TXT      A016
```

Errors detected: 0

Zorgt men nu, dat LABEL pas zijn waarde krijgt als de argumenten MESSAGE en TXT bekend zijn, dan gaat alles goed. (dus na TXT FCC 'ERROR',3 pas LABEL equ TXT-MESSAGE)

Ook een goede oplossing is:

```
A000 LDX      #TXT-MESSAGE
```

b. Bij r/w diskerror gaat as gewoon door, zonder enige melding !!!

Indien een libfile niet geopend kan worden gaat AS ook gewoon door.

Beter zou het zijn om een foutmelding te geven en te stoppen.

- c. Een optie om een bepaald deel wel te assembleren, maar niet naar de .bin file te schrijven heeft AS helaas niet. Men kan dan een assemblerprogramma als labelfile gebruiken. Volgens mij kan as het niet, ook niet met conditionele pseudos.

7. Dos-65 basic v2.00

=====

- a. KB9 Basic_2.00 Bij een Save "file" commando wordt alle data vanaf \$400 op schijf gezet, terwijl het basic werkgeheugen pas bij ca. \$2000 begint !
Na een Load "file" commando kan men deze file zondermeer RUNnen, maar als men een regel toevoegt, wordt (soms) vooral bij lange files, het programma vernietigd vanaf het nieuw ingevoerde regelnummer.
- b. Problemen met subdirectories bij 9 karakters tellende filenamen (voor de punt)
BV: LOAD"f/fruitmach" geeft foutmelding.
en SAVE"f/e23456789" slaat de filenaam op als:
e2345678x.BAS (met x=willekeurig)
OPLOSSING: file copieren van subdir naar hoofd-dir en dan laden of saven, of met asn. Dit is natuurlijk een lapmiddel.
- c. Bij output redirect > -tf file g 0 klopt de scherm-presentatie niet meer!
Bv. bij het invullen van het belastingformulier, wordt alles op de 1e regel geprint. Waarschijnlijk worden de linefeeds gefilterd ??
Cursor to werkt wel goed.

KB9 Basic_2.10 LIST 10-50 bv is onmogelijk

- d. Foutcorrectie, inserten, deleten van een regel, heeft tot gevolg dat de hele basicfile vernietigd wordt.
Indien een letter wordt ingetikt als een getal verwacht wordt geeft basic geen 'redo from start' maar een syntaxerror
Bovendien wordt het programma afgebroken !
etc. etc.

8. C

====

- a. compiler vangt ongedefinieerde labels niet. Dit komt er pas tijdens het assembleren uit, en dan zijn er al veeeeeele minuten verstreken...
- b. Input ^D van toetsenbord is afwijkend.

9. Manuals

=====

- a. In de manual DOS-65 ENTRIES van Wim Schimmel staat een fout bij het beschrijven van SOPT: (option scanner)
pagina 20: moet zijn: X=0110 0000

<>

Getallen (deel 1)

Door Gert van Opbroek
Bateweg 60
2481 AN Woubrugge

Inleiding

In deze serie wil ik beschrijven hoe een computer omgaat met getallen en hoe hij er mee rekent. In dit eerste deel worden met name de gehele getallen behandeld. In het tweede deel wil ik de drijvende-komma getallen (floating point) behandelen. In het derde en eventueel volgende deel worden onder andere de floating point coprocessors behandeld.

Soorten getallen

In de wiskunde kent men diverse verzamelingen van getallen. In de eerste plaats zijn dat de **Natuurlijke** getallen. Dit zijn de niet negatieve gehele getallen 0, 1, 2, 3, Verder zijn er de **Gehele** getallen. Dit zijn de getallen 0, -1, 1, -2, 2, -3, 3, Een derde verzameling zijn de zogenaamde **Rationele** getallen. Dit zijn alle gehele getallen uitgebreid met alle breuken dus o.a. $1/2$, $127/255$, 11 $13/17$ etc. Als vierde noem ik de **Reële** getallen. Dit zijn de rationele getallen plus de getallen die niet als een breuk te schrijven zijn. Hiervan zijn wortel 2 en het getal pi voorbeelden. Behalve de hiergenoemde verzamelingen, zijn er nog enkele, maar voor dit verhaal zijn dit de belangrijkste.

Floating Point getallen

Om twee rationele getallen bij elkaar op te tellen moet nogal wat werk verzet worden. Als voorbeeld de volgende rekensom:

$$3/4 + 1/3 = 9/12 + 4/12 = 13/12 = 1 \frac{1}{12}$$

Omdat dit, vooral bij grotere berekeningen, nogal omslachtig is, werkt men meestal met een decimale representatie:

$$3/4 = 75/100 = 0.75$$

$$1/3 = 33/100 = 0.33$$

$$3/4 + 1/3 = 0.75 + 0.33 = 1.08$$

In het bovenstaande voorbeeld is de gelijkheid:

$$1/3 = 33/100$$

niet geheel correct. Het is zelfs zo dat het getal $1/3$ helemaal niet decimaal weer te geven valt, daarvoor zouden we oneindig

veel cijfers nodig hebben. De fout die op deze manier gemaakt wordt, noemen we **Afrondingsfout**. In de praktijk heeft men het volgende afgesproken:

Bereken het decimale getal voor het aantal gewenste decimalen plus ϵ . Is de laatste decimaal 5 of hoger, verhoog dan de op ϵ na laatste decimaal met één. Voorbeelden voor twee decimalen:

$$1/3 = 0.33, \quad 1/6 = 0.67, \quad 1/8 = 0.13$$

Als we nu maar genoeg decimalen (cijfers achter de komma) gebruiken, dan is de berekening toch nog voldoende nauwkeurig.

Vooraf bij vermenigvuldigen en delen komt het nogal eens voor dat we hele grote en hele kleine getallen met elkaar moeten vermenigvuldigen of op elkaar moeten delen. Een voorbeeld:

Eén schroef weegt 1.432 gram of 0.001432 kilogram. Hoeveel wegen 200 doosjes met een gros (144) van deze schroefjes?

$$\text{Aantal schroefjes: } 200 * 144 = 28800$$

$$\text{Gewicht: } 28800 * 0.001432 = 412.416 \text{ kg}$$

Om deze reden heeft men de drijvende komma notatie ingevoerd. Hierbij wordt het getal geschreven als een product van een zogenaamde **mantissa** en een **exponent**. In het decimale stelsel gebruikt men uiteraard een decimale exponent. Als voorbeeld:

$$0.001432 = 1.432E-3$$

$$28800 = 2.8800E4$$

Dit wordt uitgesproken als ϵ komma vier drie twee maal tien tot de macht drie. Behalve de bovenstaande notatie zijn er nog een paar maar dit is de meest gebruikte notatie. Opvallend is, dat de decimale punt altijd achter het eerste cijfer staat. Het is een afspraak dat de exponent zodanig is, dat de decimale punt direct achter de eerste decimaal terechtkomt. Een getal dat aan die afspraak voldoet, noemen we **genormeerd**. Omdat de komma als het ware naar de goede positie drijft, noemen we dergelijke getallen floating point numbers (drijvende komma getallen).

Getalstelsels

Het vervolg van dit artikel gaat uitsluitend nog over gehele getallen. De uitwerking van de floating point getallen komt in deel 2 aan bod.

Als we het volgende getal nader bekijken dan valt ons iets bijzonders op:

$$\begin{aligned} 14876 &= 1 * 10000 + \\ &4 * 1000 + \\ &8 * 100 + \\ &7 * 10 + \\ &6 * 1 \end{aligned}$$

Nader uitgewerkt is dit:

$$\begin{aligned} 14876 &= 1 * 10 \text{ tot de macht } 4 + \\ &4 * 10 \text{ tot de macht } 3 + \\ &8 * 10 \text{ tot de macht } 2 + \\ &7 * 10 \text{ tot de macht } 1 + \\ &6 * 10 \text{ tot de macht } 0 \end{aligned}$$

Als men dus de posities van de cijfers in het getal van rechts naar links nummert, te beginnen met nul, dan geeft de plaats de macht van 10 waarmee het betreffende cijfer vermenigvuldigd moet worden aan.

Nu kan men zich afvragen waarom we in het dagelijks leven gebruik maken van 10-machten en dus zodoende van het tientallige of decimale stelsel. Het antwoord hierop is eigenlijk zeer simpel: omdat de mens nu eenmaal tien vingers heeft. Er zijn gebieden waar men, vooral bij maten en gewichten, lange tijd gebruik gemaakt heeft van het twaalftallige stelsel maar langzaam maar zeker gaat men steeds meer over naar het tientallige stelsel. De begrippen dozijn en gros komen uit het twaalftallige stelsel immers:

$$\begin{aligned} \text{een dozijn} &= 12 \\ \text{een gros} &= 12 \text{ dozijn} = 144 \end{aligned}$$

320 decimaal is dus 2 gros 2 dozijn 8 dus

228 in het twaalftallige stelsel

Mensen die de programmeertaal FORTH op hun systeem hebben kunnen in elk getalstelsel werken door het intypen van de opdracht:

n BASE ! waarbij n het grondtal is.

Daar een computer werkt met zogenaamde bits (binary unit), wordt binnen een computer gebruik gemaakt van het grondtal 2. Een computer werkt dus in het tweetallige of binaire stelsel. Een getal als 145

wordt dan in een computer weergegeven als:

$$\begin{aligned} 1 * 128 &= 1 * 2 \text{ tot de macht } 7 \\ 0 * 64 &= 0 * 2 \text{ tot de macht } 6 \\ 0 * 32 &= 0 * 2 \text{ tot de macht } 5 \\ 1 * 16 &= 1 * 2 \text{ tot de macht } 4 \\ 0 * 8 &= 0 * 2 \text{ tot de macht } 3 \\ 0 * 4 &= 0 * 2 \text{ tot de macht } 2 \\ 0 * 2 &= 0 * 2 \text{ tot de macht } 1 \\ 1 * 1 &= 1 * 2 \text{ tot de macht } 0 \end{aligned}$$

Dus:

$$145 (10) = 10010001 (2)$$

Omdat lange reeksen van enen en nullen lastig te lezen zijn, worden in de praktijk vaak vier bits samen genomen. In dit geval werken we dan in het hexadecimale (zestientallige) stelsel. Op sommige computers (PDP-11) werkt men in het octale (achtentallige) stelsel. In ons voorbeeld:

$$145 (10) = 10010001 (2) = 91 (16)$$

Uit dit voorbeeld is waarschijnlijk wel duidelijk, dat het noodzakelijk is dat het grondtal aangegeven wordt. In het zestientallige stelsel wordt dit wel gedaan door een '\$' voor het getal te zetten. In het binaire stelsel wordt wel een '%' voor het getal gezet.

Binary Coded Decimal

Behalve de binaire getallen zijn er ook computers die gebruik maken van een andere representatievorm. Dit is de zogenaamde Binary Coded Decimal of BCD-notatie. De 6502 kan ook van deze representatie gebruik maken. In de BCD-notatie wordt uitgegaan van het tientallige stelsel. Elke decimaal wordt dan weergegeven met een bitpatroon van vier bits:

$$\begin{aligned} 0 &= 0000 \\ 1 &= 0001 \\ 2 &= 0010 \\ &\dots \dots \dots \\ 8 &= 1000 \\ 9 &= 1001 \end{aligned}$$

Het voordeel van deze notatie is gelegen in het gemak waarmee de conversie van en naar decimale getallen gemaakt kan worden. Nadelen zijn het feit dat de rekenalgoritmen moeilijker zijn dan bij een binaire representatie en het feit dat er ongeveer 30 % meer bits voor nodig zijn om een getal weer te geven. BCD wordt vooral gebruikt op de IBM mainframes. Deze machines worden met name gebruikt voor administratieve toepassingen en juist daar speelt het gemak van de conversie een

belangrijke rol.

Het optellen in het binaire stelsel

Hoe men in het decimale stelsel moet rekenen, is iedereen wel bekend. Hoe gaat optellen nu in zijn werk in het binaire stelsel? Stel we willen de (decimale) getallen 21 en 15 bij elkaar optellen:

$$\begin{array}{r} 21 = 10101 \\ 15 = 01111 \end{array}$$

De bewerking gaat nu van achter naar voren als volgt:

$$\begin{array}{rcl} 1 + 1 & = & 0 \quad \text{en 1 onthouden} \\ 1 + 0 + 1 & = & 0 \quad \text{en 1 onthouden} \\ 1 + 1 + 1 & = & 1 \quad \text{en 1 onthouden} \\ 1 + 0 + 1 & = & 0 \quad \text{en 1 onthouden} \\ 1 + 1 + 0 & = & 0 \quad \text{en 1 onthouden} \\ 1 + 0 + 0 & = & 1 \end{array}$$

De uitkomst is dus $100100 = 36$.

Opvallend is, dat we bij het optellen van twee getallen met vijf bits een getal met zes bits als resultaat krijgen. Dit is in het algemeen zo, het optellen van twee getallen van n bits levert als resultaat een getal van maximaal $n+1$ bits. Dit kan op een computer die met acht bits rekent tot resultaat hebben dat we een getal met negen bits als antwoord krijgen. Dit bit kunnen we echter niet bij de laagste acht bits opslaan. Als oplossing hiervoor heeft men de CARRY vlag ingevoerd. In deze vlag wordt dit negende bit opgeslagen en deze kan dan automatisch weer bij de volgende optelling meegegenomen worden (zoals één onthouden bij een optelling). Op deze manier kan men met een acht bits computer ook grotere getallen gebruiken. In de praktijk worden op acht bits computers meestal getallen van 15 bits of 16 bits gebruikt. Bij een 15 bits getal heeft het voorste bit een andere functie, ik kom daar nog op terug.

Positieve en negatieve getallen

Als we uitsluitend met positieve gehele getallen werken, dan kunnen we alle bits van het getal gebruiken. In dit geval spreken we van zogenaamde **Unsigned Numbers**. Als we zestien bits voor een dergelijk getal reserveren, dan kunnen we dus getallen van 0 tot en met 65535 gebruiken. In de praktijk willen we echter ook vaak met negatieve getallen werken. Om deze getallen ook in het binaire stelsel uit te kunnen drukken zijn een viertal verschillende mogelijkheden aanwezig:

1: Signed magnitude:

Het voorste bit van het getal geeft aan of het getal positief of negatief is. Dit is dus eigenlijk hetzelfde als in het decimale stelsel waarbij we een '-' teken voor het getal plaatsen.

Voorbeelden:

$$\begin{array}{rcl} 0 & = & 0000 \ 0000 \ 0000 \ 0000 \\ 1 & = & 0000 \ 0000 \ 0000 \ 0000 \\ -1 & = & 1000 \ 0000 \ 0000 \ 0000 \\ 32767 & = & 0111 \ 1111 \ 1111 \ 1111 \\ -32767 & = & 1111 \ 1111 \ 1111 \ 1111 \end{array}$$

Bij signed magnitude hebben we twee representaties voor het getal 0. Een positieve versie:

$$0000 \ 0000 \ 0000 \ 0000$$

en een negatieve versie:

$$1000 \ 0000 \ 0000 \ 0000$$

2: Excess:

In dit geval wordt nul niet gerepresenteerd door allemaal nullen maar door een ander bitpatroon. De overige getallen worden dan ten opzichte van dit binaire getal uitgerekend.

Voorbeelden in de excess 32768-notatie:

$$\begin{array}{rcl} 0 & = & 1000 \ 0000 \ 0000 \ 0000 \\ 1 & = & 1000 \ 0000 \ 0000 \ 0001 \\ -1 & = & 0111 \ 1111 \ 1111 \ 1111 \\ 32767 & = & 1111 \ 1111 \ 1111 \ 1111 \\ -32767 & = & 0000 \ 0000 \ 0000 \ 0001 \end{array}$$

In dit voorbeeld is het getal dus het volledige zestien bits getal - 32768. In de excess-notatie zijn er meer negatieve getallen dan positieve.

3: 1's complement (one's complement):

In deze representatie krijgt men van een positief getal de negatieve versie door van alle bits het complement te nemen (dus een 1 wordt een 0 en andersom).

Voorbeelden:

$$\begin{array}{rcl} 0 & = & 0000 \ 0000 \ 0000 \ 0000 \\ 1 & = & 0000 \ 0000 \ 0000 \ 0001 \\ -1 & = & 1111 \ 1111 \ 1111 \ 1110 \\ 32767 & = & 0111 \ 1111 \ 1111 \ 1111 \\ -32767 & = & 1000 \ 0000 \ 0000 \ 0000 \end{array}$$

Bij 1's complement geeft het hoogste bit altijd het teken van het getal aan. Ook

bij 1's complement hebben we een positieve en een negatieve versie van het getal 0:

```
0000 0000 0000 0000
1111 1111 1111 1111
```

4: 2's complement (two's complement):

Bij 2's complement krijgt men de negatieve versie van een getal door eerst het 1's complement te nemen en dan bij het binaire getal één op te tellen.

Voorbeelden:

```
0 = 0000 0000 0000 0000
1 = 0000 0000 0000 0001
- 1 = 1111 1111 1111 1111
32767 = 0111 1111 1111 1111
- 32767 = 1000 0000 0000 0001
```

Deze representatie kent dus, evenals de excess-notatie slechts één, positieve nul. Maar ook meer negatieve dan positieve getallen.

Natuurlijk heeft elke representatie zijn specifieke voor- en nadelen. Laten we eens een simpele optelling in alle notaties bekijken:

$$- 1 + 2 = 1$$

Signed Magnitude:

$$1000\ 0001 + 0000\ 0010 = 0000\ 0001$$

In dit geval moeten we, evenals bij de decimale notatie het voorbeeld omwerken naar $2 - 1 = 1$.

Excess 128:

$$0111\ 1111 + 1000\ 0010 = 1000\ 0001$$

Bij excess vraagt het voorste bit speciale aandacht. Men moet altijd het getal nul = 1000 0000 bij het resultaat optellen. De overige bits verkrijgt men door eenvoudige recht-toe recht-aan bewerking.

1's complement:

$$1111\ 1110 + 0000\ 0010 = 0000\ 0001$$

In dit geval wordt de carry die uit de optelling van het voorste bit komt nog eens bij het resultaat opgeteld. Deze bewerking wordt carry around genoemd.

2's complement:

$$1111\ 1111 + 0000\ 0010 = 0000\ 0001$$

Bij 2's complement is optellen van positieve en negatieve getallen recht-toe recht-aan en geeft, ook bij tekenwisseling, geen complicaties. Op bijna alle hedendaagse computers wordt voor de representatie van gehele getallen gebruik gemaakt van deze notatie. Voor de verdere beschrijving van de gehele getallen (signed numbers) ga ik uit van de 2's complement notatie.

Het aftrekken van gehele getallen

Als we twee gehele getallen van elkaar af willen trekken, dan wordt over het algemeen de negatieve waarde van de tweede operand bij de eerste opgeteld. De meeste computers hebben wel een aftrek-instructie maar in de praktijk zal de processor intern deze operatie uitvoeren. De carry is in dat geval een zogenaamde borrow. Dit wil zeggen dat het resultaat van de aftrekking nog met 1 verlaagd moet worden. Bij de 6502 geeft de carry de inverse van het borrow bit weer; heeft men geen borrow, dan moet de carry-vlag gezet zijn.

Overflow

Als we twee relatief grote getallen bij elkaar optellen, dan kan het voorkomen dat we inplaats van een positief getal een negatief getal krijgen.

Voorbeeld:

$$20000 + 30000 =$$

$$0100\ 1110\ 0010\ 0000 + 0111\ 0101\ 0011\ 0000 \\ = 1100\ 0011\ 0101\ 0000$$

Deze situatie noemt men **Overflow**. Eenzelfde situatie kan voorkomen als men twee relatief grote negatieve getallen bij elkaar optelt:

$$- 20000 + - 30000 =$$

$$1011\ 0001\ 1110\ 0000 + 1000\ 1010\ 1101\ 0000 \\ = 0011\ 1100\ 1011\ 0000$$

Deze situatie kan worden herkend aan het volgende feit:

Als de carry van de optelling ongelijk is aan de carry bij de optelling van het tweede bit van links. Dus als de carry naar het tekenbit ongelijk is aan de carry uit het tekenbit.

Een overflowconditie wordt over het algemeen door de processor herkend en aangege-

ven met een overflowvlag.

Rekenalgoritmen

De algorithmen voor het optellen en aftrekken zijn reeds behandeld. Voor de volledigheid nog twee korte stukjes 6502-code voor deze operaties op 16 bits getallen:

```
CLC          ; Clear carry flag
CLD          ; Binary mode
LDA  ADR1    ; LSB of operand 1
ADC  ADR2    ; Add LSB of operand 2
STA  ADR1    ; LSB of result
LDA  ADR1-1  ; MSB of operand 1
ADC  ADR2-1  ; Add MSB of operand 2
STA  ADR1-1  ; MSB of result
```

In het bovenstaande voorbeeld wordt operand 2 bij operand 1 opgeteld. Na de optelling van het MSB geeft de overflow-vlag aan of er een overflow-conditie opgetreden is. Met een BVS kan men dus eventueel een foutafwikkeling activeren.

```
SEC          ; Set carry flag
CLD          ; Binary mode
LDA  ADR1    ; LSB of operand 1
SBC  ADR2    ; Subtract LSB 2
STA  ADR1    ; LSB of result
LDA  ADR1-1  ; MSB of operand 1
ADC  ADR2-1  ; Subtract MSB 2
STA  ADR1-1  ; MSB of result
```

Ook in dit geval wordt bij een overflow-conditie de overflow-vlag gezet.

Bij het vermenigvuldigen moet men zich realiseren dat het totaal aantal bits gelijk kan worden aan de som van het aantal bits van in de operanden. Dit wil dus zeggen dat bij de vermenigvuldiging van twee 8 bits getallen een zestien bits getal kan ontstaan. Het vermenigvuldigen van binaire getallen gaat op dezelfde manier als bij decimale getallen:

```
  123
  64
---- *
  492
 738
---- +
 7872
```

De tussenresultaten 492 en 738 heten partieel product. In het binaire stelsel gaat

het op dezelfde manier:

```
  0111
  0101
----- *
  0111
 0000
 0111
 0000
----- +
0100011
```

Het algoritme is eenvoudig: is het bit dat men onder behandeling heeft een 1, dan is het partieel product de eerste operand, is dit bit 0, dan is het partieel product nul. Een zeer fraai (en snel) algoritme voor de vermenigvuldiging van twee 8 bits getallen staat beschreven in het boek:

Programming the 6502

van Rodney Zaks: Het product wordt met nullen gevuld. Bij iedere vermenigvuldiging wordt het partiele product bij het resultaat opgeteld. Hierbij werkt men steeds op het MSB. Aansluitend wordt het complete 16 bits resultaat een positie naar rechts geschoven.

```
MULT LDA  #0          ; Initialize result
      STA  B          ; Result LSB
      LDX  #8          ; Loop index
LOOP  LSR  C          ; Shift operand 2
      BCC  NOAD        ; Bit set ?
      ADC  D          ; Add operand 1
NOAD  ROR  A          ; Shift result
      ROR  B          ; Catch bit in LSB
      DEX             ; Decrement index
      BNE  LOOP        ; Ready ?
```

De twee operanden staan resp. in de geheugenlocaties D en C. Het resultaat komt in de accu (MSB) en in de geheugen-locatie B.

Helaas kan dit algoritme niet gebruikt worden voor negatieve getallen. Bij negatieve getallen moet men net zo handelen als bij de decimale vermenigvuldiging: eerst de operanden positief maken, is één en slechts één van de operanden negatief, dan moet men ook het resultaat negatief maken.

Ook het delen lijkt heel veel op een deci-

male deling:

```

12 /7423\ 618
  72
  --
  22
  12
  --
  103
  96
  ---
   7
    
```

Binair:

```

0011 / 10011011\ 00011011
  11
  --
  11
  11
  --
   1
   0
   -
  10
   0
  --
  101
  11
  ---
  101
  11
  ---
   10
    
```

Ook dit algoritme werkt helaas niet voor negatieve getallen. Uit

Programming the 6502

komt de volgende routine:

```

DIV LDY #8          ; Loop index
SEC
SBC D              ; Trek de deler af
LOOP PHP          ; Onthoud de vlaggen
ROL Q             ; Schuif het resultaat
ASL B             ; LSB
ROL A             ; Schuif tussenres.
PLP              ; Haal vlaggen
BCC ADD          ; Deler groter ?
SBC D            ; Nee, trek af
JMP NEXT        ; volgend bit
ADD ADC D        ; Tel deler op
NEXT DEY         ; Verlaag index
BNE LOOP
BCS LAST        ; Laatste bew. OK
ADC D           ; Tel deler op
CLC
LAST ROL Q      ; Corrigeer rest
BRK            ; einde
    
```

In deze routine staat het deeltal in de Accu (MSB) en de locatie B (LSB). Ook het

resultaat (quotient) komt in de accu en in B terecht. De deler staat in D. Q wordt gebruikt voor de tussenresultaten. Bovendien bevat Q, na afloop, de rest van de deling.

De getoonde routine werkt niet helemaal zoals in de getoonde staartdeling. Als een bit in het quotient nul wordt, wordt er bij de volgende bewerking niet afgetrokken maar opgeteld. Hiermee wordt net zolang doorgedaan totdat het tussenresultaat weer een carry geeft. Dit betekent dat het actuele bit weer één is. Al met al een leuke puzzel om te onderzoeken of het echt klopt. Als aanwijzing voor de puzzelaars nog dat een schuifoperatie vermenigvuldigen met 2 (links) of delen door 2 inhoudt. Ik ben ervan overtuigd dat de routine correct is.

Samenvatting.

Meestal werkt een microprocessor met gehele getallen van zestien bits. Bij de moderne zestien bits processoren wordt ook wel gewerkt met gehele getallen van 32 bits. Een zestien bits geheel getal wordt een INTEGER2 of een SHORT genoemd, een 32 bits geheel getal een INTEGER4 of een LONG.

Alle microprocessors werken met de binaire notatie van getallen, enkele, waaronder de 6502, kunnen bovendien met BCD-notatie werken.

De gehele getallen zijn op een computer onder te verdelen in UNSIGNED getallen die alle bits voor het positieve getal kunnen gebruiken en in SIGNED getallen waarbij ook de negatieve getallen geïmplementeerd zijn. Voor de SIGNED gehele getallen gebruiken bij mijn weten alle microprocessors de 2's complement notatie.

In deel 2.

In deel twee van deze serie worden de floating point getallen behandeld. Hierbij wordt de IEEE (aai triple ie) -definitie als voorbeeld gebruikt. De in dit deel genoemde signed magnitude en excess notatie voor negatieve getallen zullen dan weer aan de orde komen.

Als er niets mis gaat, zal deel twee geplaatst worden in het decembernummer van de 6502 Kenner.

=====

DOS65 Game Library

```

1 ;*****
2 ;*
3 ;* DEMO V.1.88 a Demonstration Video Game for the DOS65 System *
4 ;* ----- *
5 ;*
6 ;* Written by A.G.Megens June 1988 *
7 ;* Haringvliet 371 *
8 ;* 8032 HK Zwolle phone: 038-537073 *
9 ;*
10 ;* This program shows how to use the GAME library files *
11 ;* Feel free to change this program as much as you want, some *
12 ;* routines definitely can use some improvements! *
13 ;* e.g. add an instrument panel on the screen with alltitude, *
14 ;* fuel and bombs left etc.etc. *
15 ;*****
16 opt nogen
17 ttl 'tp.DEMO GAME V.1.88 for DOS65 %s page %02
18 pag 85
19 ORG $5000
20
21 DEMO JMP START
22
23 LIB JMPVAR ;jump table and variables
24
25 RBDIST EQU 5 ;minimum bomb distance
26 ;(increase it to make the game more difficult)
27 ;(do not make it less than 2 for a bomb is 2)
28 ;(characters in height)
29 START LDA #3
30 STA RND ;init Random Number Generator
31 STA RND+1 ;(lib. file RANDOM.MAC)
32 STA RND+2
33 ;
34 TITLE JSR CLS ;clear screen (lib. file SCREEN.MAC)
35 JSR PRINT ;print title screen
36 fcc TAB,TAB,TAB,'D E M O N S T R A T I O N',CR
37 fcc TAB,TAB,TAB,' G A M E',CR
38 fcc TAB,TAB,TAB,' U S I N G',CR
39 fcc TAB,TAB,TAB,' T H E',CR
40 fcc TAB,TAB,TAB,' G A M E',CR
41 fcc TAB,TAB,TAB,' L I B R A R Y',CR,0
42 JSR INVERS ;inverse video (lib.file SCREEN.MAC)
43 JSR PRINT
44 fcc CR,CR
45 fcc ' for DOS65 by A.Megens
46 fcc CR,CR,0
47 JSR NORMAL ;normal video (lib.file SCREEN.MAC)
48 JSR PRINT ;print menu
49 fcc ESC,'iI',ESC,'nnstructions'
50 fcc TAB,ESC,'iC',ESC,'nlock speed'
51 fcc TAB,ESC,'iD',ESC,'nefine keys'
52 fcc TAB,ESC,'iS',ESC,'ntart game'
53 fcc TAB,ESC,'iR',ESC,'return to DOS',CR,CR,0
54 JSR HITABLE ;show high scores table (lib.file SCORE.MAC)
55 MENU JSR WFKEY ;wait for key (lib.file KEY.MAC)
56 AND #%01011111 ;make uppercase
57 CMP #'I'
58 BNE DKEY
59 JSR INSTRUC ;show instructions
60 JMP TITLE

```

```

61 DKEY    CMP    #'D'
62         BNE    CKEY
63         JSR    DEFINE ;define new keys (lib.file KEY.MAC)
64         JMP    TITLE
65 CKEY    CMP    #'C'
66         BNE    SKEY
67         JSR    CLOCK ;clock speed (lib.file SYSCLK.MAC)
68         JSR    RTITLE
69         JMP    TITLE
70 SKEY    CMP    #'S'
71         BNE    RKEY
72         LDA    #100
73         STA    PAUSE ;initial speed
74         LDA    #1
75         STA    ROUND
76         LDA    #3
77         STA    LIVES
78         JMP    SGAME ;start game
79 RKEY    CMP    #'R'
80         BNE    MENU ;ignore other keys
81         JMP    DOS ;return to DOS65 (lib.file ENDGAME.MAC)
82 ;=====
83 ; entry for next round
84 ;=====
85 NEXT    JSR    PRINT
86         fcc    $14,25,1,'Press space-bar to continue.....',BELL,0
87 WCONT    JSR    WAIT ;wait loop (lib.file WAIT.MAC)
88         JSR    INKEY
89         CMP    QUIT
90         BNE    1.f
91         JMP    TITLE
92 1        CMP    #' '
93         BNE    WCONT
94         JSR    PRINT
95         fcc    $14,25,1,' ',0
96 ;=====
97 ; game starts here
98 ;=====
99 SGAME    JSR    CLS
100         JSR    INVERS
101         JSR    PRINT
102         fcc    $14,1,25,
103         fcc    'Time:          Round:          Score:          0          High score:
0 Planes:'
104         fcc    0
105         JSR    NORMAL
106         LDX    #0
107         TXA
108         STA    GEAROUT
109         STA    LANDF
110 CLRBOMB STA    BOMBY,X ;clear bombs
111         INX
112         CPX    #10
113         BNE    CLRBOMB
114         JSR    UPDATE ;update GAME status line
115         LDA    #1 ;start position of plane
116         STA    PLANEX
117         STA    PLANEY
118         STA    BDIST
119         JSR    DRPLANE
120         JSR    DRSURF ;draw surface
121 ;=====
122 ; MAIN LOOP
123 ;=====

```

```

124 MAINL INC COUNT
125 JSR INPUT ;check user input
126 BMI QMAIN ;QUIT key depressed, return to title
127 BCS 1.f ;a key was depressed
128 DEC BDIST ;else decr. bomb distance
129 BNE 1.f ;and continue if <>0
130 LDA #1 ;else reset it to 1
131 STA BDIST
132 1 JSR DRBOMBS ;draw bombs
133 JSR DRPLANE ;draw plane
134 JSR WAIT ;slow down
135 JSR ERPLANE ;erase plane
136 JSR ERBOMBS ;erase bombs
137 JSR MVPLANE ;move plane
138 BMI CRASHED ;it crashed!
139 BCC NXTRND ;landed!
140 JSR MVBOMBS ;move bombs and check for hits
141 JMP MAINL
142 QMAIN JMP TITLE ;QUIT to title screen
143 NXTRND LDA ROUND ;goto next round
144 CLC
145 SED ;decimal number!
146 ADC #1
147 CLD
148 STA ROUND
149 LDA PAUSE ;faster this time
150 SEC
151 SBC #10
152 CMP #20
153 BCS 1.f
154 LDA #20
155 1 STA PAUSE
156 JMP NEXT
157 CRASHED JSR EXPLODE ;the Turkey crashed!
158 DEC LIVES ;get one of his lives
159 BEQ GOVER ;well that's it, he's out
160 JMP NEXT ;give it another go
161 GOVER JMP ENDGAME ;did he at least improve the high score?
162 ;=====
163 ; Routine for FIRE key
164 ;=====
165 DROPB DEC BDIST ;previous bomb far enough?
166 BNE FEX ;no, ignore FIRE key
167 LDA PREVKEY ;get previous key
168 CMP FIRE ;has FIRE key been released?
169 BEQ 5.f ;no, then ignore FIRE key
170 LDX #0 ;room in table?
171 FLOOP LDA BOMBY,X
172 BNE 1.f ;try next
173 LDA #12
174 STA ASAV
175 LDA PLANEX ;there is room, put coordinates in table
176 JSR OFFSET
177 CMP #79
178 BEQ 3.f ;too far right to drop bomb
179 STA BOMBX,X
180 LDA PLANEY
181 CLC
182 ADC #4
183 CMP #23 ;too low for another bomb
184 BCS 3.f
185 STA BOMBY,X
186 3 LDA #RBDIST ;reset bomb distance
187 STA BDIST

```

```

188      RTS
189  1      INX
190      CPX      #10      ;place for 10 bombs in table
191      BNE      FLOOP
192  5      LDA      #1
193      STA      BDIST
194  FEX      LDA      #0
195      RTS
196  ;=====
197  ; routine for GEAR key
198  ;=====
199  MGEAR      LDA      GEAROUT
200      EOR      #$FF
201      STA      GEAROUT
202      LDA      #0
203      RTS
204  ;=====
205  ; routine for LAND key
206  ;=====
207  LANDIT      LDA      #1
208      STA      LANDF
209      RTS
210  ;-----
211  ; Game dependant routine, copied from SHWKEY.MAC file
212  ; then changed to match this game
213  ;-----
214  SHWKEY      JSR      PRINT
215      fcc      $14,25,10,"Drop bomb.....",ESC,"i",0
216      LDA      FIRE
217      JSR      SPECIAL
218      JSR      PRINT
219      fcc      ESC,"n"
220      fcc      $14,25,12,"Landing gear.....",ESC,"i",0
221      LDA      GEAR
222      JSR      SPECIAL
223      JSR      PRINT
224      fcc      ESC,"n"
225      fcc      $14,25,14,"Land plane.....",ESC,"i",0
226      LDA      LAND
227      JSR      SPECIAL
228      JSR      PRINT
229      fcc      ESC,"n"
230      fcc      $14,25,16,"Quit to main menu...",ESC,"i",0
231      LDA      QUIT
232      JSR      SPECIAL
233      JSR      PRINT
234      fcc      ESC,"n"
235      fcc      $14,25,18,"Halt/Pause key.....",ESC,"i",0
236      LDA      HALTKEY
237      JSR      SPECIAL
238      JSR      PRINT
239      fcc      ESC,"n",0
240      RTS
241  ;-----
242  ; Game dependant routine, copied from INSTRUC.MAC file
243  ; then changed to match this game
244  ;-----
245  INSTRUC      JSR      CLS
246      JSR      PRINT
247      fcc      "Bomb the surface to get if flat enough for your landing.",CR
248      fcc      "The bomb reload mechanism takes some time to reload, so if th
e FIRE",CR
249      fcc      "key is not responding, the next bomb is not loaded yet!",CR
250      fcc      "The mechanism gets jammed if you do not release the FIRE key."

```



```

251      ^,CR      fcc      ^The following control keys can be used: ^,0
252      JSR      SHWKEY
253      RTITLE   JSR      WFUSER
254      RTS
255      ;
256      ;          DRSURF  Draw random surface, height depends on
257      ;          ;
258      ;          ;
259      DRSURF   JSR      INVERS      ;draw inverse spaces
260      LDX      #1
261      1        STX      XSAV
262      LDA      ROUND      ;get current round
263      CLC
264      ADC      #5          ;and add an offset of 5
265      STA      ASAV
266      2        JSR      RANDOM      ;get me a number, any number!
267      AND      #$1F        ;throw away some bits
268      ADC      #3          ;add 3 (to get rid of NULL)
269      CMP      ASAV        ;it is still small enough?
270      BCS      2.b         ;no, this one is garbage, try again
271      STA      ASAV        ;perfect!, use it to draw one height
272      LDA      #24        ;starting at line 24
273      SEC
274      SBC      ASAV        ;and up the computed height
275      LDX      XSAV
276      STA      SURFACE,X   ;also store it for crash checking
277      LDY      #24        ;now draw the little bugger
278      STY      YPOS
279      STX      XPOS
280      3        JSR      GOTOXY      ;goto X,24
281      JSR      PRINT      ;and print one (inverted) space
282      fcc      ^ ^,0
283      DEC      YPOS        ;go up one line
284      DEC      ASAV        ;decr. height counter
285      BNE      3.b         ;still not there, keep going
286      INX
287      CPX      #80        ;ok, we're done, goto next height
288      BNE      1.b         ;complete bottom line done?
289      JSR      NORMAL      ;(sigh) not yet, do it again
290      RTS              ;hey it worked, just reset video
291      ;              ;and we're done!
292      ;          ;
293      ;          ;
294      ;          X,Y
295      ;          01234567890123456789012
296      ;          0
297      ;          1
298      ;          2
299      ;          3
300      ;          4
301      ;          ;
302      DRPLANE  LDX      PLANEX      ;put cursor where the plane is
303      LDY      PLANEY
304      STX      XPOS
305      STY      YPOS
306      LDX      #3          ;plane sprite is number 3 in SORT ta
307      JSR      PLOT        ;plot it
308      LDA      GEAROUT      ;check if gear is out
309      BEQ      1.f         ;no, then we're done
310      LDA      PLANEY      ;compute position of gear
311      CLC          ;3 lines down
312      ADC      #3
313      STA      YPOS

```

```

314          LDA      #14          ;and 14 char. in X-direction
315          STA      ASAV
316          LDA      PLANEX
317          JSR      OFFSET        ;compute X (with wrap around)
318          STA      XPOS          ;got it
319          LDX      #4            ;gear sprite is number 4
320          JSR      PLOT          ;plot it
321      1      RTS
322      ;
323      ; Erase plane, similar routine except we call ERASE here
324      ;
325      ERPLANE LDX      PLANEX
326              LDY      PLANEY
327              STX      XPOS
328              STY      YPOS
329              LDX      #3
330              JSR      ERASE
331              LDA      GEAROUT    ;check if gear is out
332              BEQ      1.f        ;no, then we're done
333              LDA      PLANEY
334              CLC
335              ADC      #3
336              STA      YPOS
337              LDA      #14
338              STA      ASAV
339              LDA      PLANEX    ;else erase that too
340              JSR      OFFSET
341              STA      XPOS
342              LDX      #4        ;gear sprite
343              JSR      ERASE
344      1      RTS
345      ;
346      ; Let plane explode (show 3 explosion sprites at plane position)
347      ;
348      EXPLODE LDA      PAUSE
349              STA      PSAV
350              LDA      #0
351              STA      PAUSE
352              LDX      PLANEX    ;put cursor where the plane is
353              LDY      PLANEY
354              STX      XPOS
355              STY      YPOS
356              LDX      #0        ;explosion sprite 1
357              JSR      PLOT      ;plot it
358              JSR      WAIT
359              LDX      PLANEX    ;put cursor where the plane is
360              LDY      PLANEY
361              STX      XPOS
362              STY      YPOS
363              LDX      #1        ;explosion sprite 2
364              JSR      PLOT      ;plot it
365              JSR      WAIT
366              LDX      PLANEX    ;put cursor where the plane is
367              LDY      PLANEY
368              STX      XPOS
369              STY      YPOS
370              LDX      #2        ;explosion sprite 3
371              JSR      PLOT      ;plot it
372              JSR      WAIT
373              LDX      PLANEX    ;put cursor where the plane is
374              LDY      PLANEY
375              STX      XPOS
376              STY      YPOS
377              LDX      #2        ;explosion sprite 3

```

```

378      JSR      ERASE      ;erase it
379      LDA      PSAV
380      STA      PAUSE
381      RTS
382      ;
383      ; Draw all bombs
384      ;
385      DRBOMBS LDX      #0
386      1      STX      XSAV
387      LDA      BOMBX,X
388      LDY      BOMBY,X
389      BEQ      DNB          ;no bomb
390      BMI      DREXB        ;bomb is marked for explosion, show it
391      STA      XPOS
392      STY      YPOS
393      LDX      #5          ;plot bomb
394      JSR      PLOT
395      DNB      LDX      XSAV
396      INX
397      CPX      #10          ;max. 10 bombs at the same time
398      BNE      1.b
399      3      RTS
400      DREXB  STA      XPOS          ;draw explosion
401      TYA
402      AND      #$7F          ;strip minus sign
403      STA      YPOS
404      LDX      #6          ;explosion sprite
405      JSR      PLOT
406      JMP      DNB
407      ;
408      ; Erase all bombs
409      ;
410      ERBOMBS LDX      #0
411      1      STX      XSAV
412      LDY      BOMBY,X
413      BEQ      ENB          ;no bomb here
414      BMI      4.f          ;bomb is exploded
415      LDA      BOMBX,X
416      STA      XPOS
417      STY      YPOS
418      LDX      #5
419      JSR      ERASE
420      ENB      LDX      XSAV
421      INX
422      CPX      #10
423      BNE      1.b
424      3      RTS
425      4      TYA
426      AND      #$7F          ;strip minus sign
427      STA      YPOS
428      LDA      BOMBX,X
429      STA      XPOS
430      LDX      #6          ;erase explosion sprite
431      JSR      ERASE
432      LDX      XSAV
433      LDA      #0          ;now disable bomb
434      STA      BOMBY,X
435      BEQ      ENB
436      ;
437      ; Move bombs and check for hits
438      ; (in this game they can only hit bottom)
439      ;
440      MVBOMBS LDX      #0
441      1      STX      XSAV

```

```

442      LDY      BOMBY,X
443      BEQ      2.f      ;no bomb
444      BMI      2.f      ;marked for explosion
445      INY
446      TYA      ;new position
447      STA      BOMBY,X
448      JSR      BCHECK      ;check if it hit something
449      BCC      2.f
450      LDA      ROUND
451      JSR      UPDSOR
452  2      LDX      XSAV
453      INX
454      CPX      #10
455      BNE      1.b
456  3      RTS
457      ;
458      ; Check if bomb X hit something
459      ;
460  BCHECK  LDY      BOMBY,X      ;get bomb Y coordinate
461          CPY      #23      ;hit bottom?
462          BCS      BHIT      ;yes!
463  NHIT    CLC      ;no hit
464          RTS
465  BHIT    LDA      #0      ;clear surface where bomb erased it
466          LDY      BOMBX,X
467          STA      SURFACE,Y
468          STA      SURFACE+1,Y
469          LDA      BOMBY,X
470          ORA      #$80      ;negative Y = explosion enabled
471          STA      BOMBY,X
472          SEC
473          RTS
474      ;-----
475      ; Move plane and check if it:
476      ;
477      ;           a> landed with gear out           return value
478      ;           b> landed without gear or crashed to surface A<0, C=1
479      ;           c> airborne with no problems       A=0, C=1
480      ;-----
481  MVPLANE LDX      PLANEX      ;get X position
482          INX      ;increase it
483          CPX      #80      ;reached right edge of screen?
484          BNE      1.f      ;no, keep going
485          LDA      PLANEY
486          CMP      #20      ;reached bottom?
487          BEQ      12.f      ;yes, then test gear
488          INC      PLANEY      ;else goto next line
489          LDX      #1      ;and start left again
490  1      STX      PLANEX
491          LDA      PLANEY      ;get Y-pos of plane
492          CMP      #20      ;at bottom?
493  12      BEQ      TSTGEAR      ;yes, don't go lower, test if gear is out
494          LDA      LANDF      ;is landing requested?
495          BEQ      2.f      ;no, continue
496          LDA      COUNT      ;else move plane also down
497          AND      #3      ;for every 3th call
498          BNE      2.f
499          INC      PLANEY      ;else go one line lower
500          JMP      2.f
501      ;
502  TSTGEAR LDA      GEAROUT      ;is gear out at bottom line?
503          BNE      LANDED      ;can this baby fly or what? busy with landing!
504  CRASH   LDA      #-1      ;What a Turkey!
505          SEC

```



```

0 1234567890123456789012
1 | - \ / / / / / / / / / / | o
2 | / / / / / / / / / / / / | *
3 | - / / / / / / / / / / / / |
4 | * (only c

```

```

570          STA      ASAV
571          LDA      PLANEX
572          JSR      OFFSET      ;for wrap around
573          TXA
574          LDA      YPOS      ;X-pos (+)
575          STA      YSAV
576          LDA      SURFACE,X  ;get surface heighth there
577          BEQ      3.f        ;already gone, no problem
578          CMP      YSAV      ;check wheel height
579          BCC      21.b       ;oh, no, I'm afraid it's fatal!
580  3          LDA      #0      ;hey, what a pilot!
581          SEC
582          RTS      ;no problem
583
584  ;
585  ; OFFSET Compute X coordinate with wrap around
586  ; Offset is in ASAV, A contains start X
587  ; YPOS is start Y
588  OFFSET CLC
589          ADC      ASAV      ;add offset
590          CMP      #80      ;wrap around?
591          BCC      2.f        ;no
592          SEC      ;else decr. X-pos
593          SBC      #79
594          INC      YPOS      ;and go one line lower
595  2          RTS
596
597  ;-----
598  ; The sprite tables contain the graphic data for all objects
599  ; the data may contain ESC sequences to get the display in the
600  ; right mode first. The ^A or <1> at the end of each line causes
601  ; the YPOS pointer to go to the next line, NULL means end of data
602  ;-----
603  SPRITE equ $ ;first sprite table
604
605  EXPLOS1 fcc ^ \ \ | / \ \ | / \ \ | / ^ ,1
606          fcc ^ - *** - *** - *** - ^ ,1
607          fcc ^ / | \ \ / | \ \ / | \ \ ,0
608
609  EXPLOS2 fcc ^ \ \ | / ^ ,1
610          fcc ^ - * - ^ ,1
611          fcc ^ / | \ \ ^ ,0
612
613  EXPLOS3 fcc ^ \ \ | / ^ ,1
614          fcc ^ - * - ^ ,1
615          fcc ^ / | \ \ ^ ,0
616
617  SPRITE2 equ $ ;use SPRITE2 if SPRITE table > 25
618
619  PLANE fcc ^ | ^ \ \ / \ \ / \ \ / \ \ / | ^ ,1
620          fcc ^ | / \ / \ \ / \ \ / \ \ / | ^ ,1
621          fcc ^ | / \ / \ \ / \ \ / \ \ / | ^ ,1
622          fcc ^ | / \ / \ \ / \ \ / \ \ / | ^ ,1
623
624  LGEAR fcc ^ | / ^ ,1
625          fcc ^ 0 ^ ,0
626
627  BOMBS fcc ^ \ \ / ^ ,1
628          fcc ^ ( ) ^ ,0
629
630  EXPLOS fcc ^ \ \ / ^ ,1
631          fcc ^ / \ \ ^ ,0
632
633  ;
634  ; contains offset in sprite table(s)
635  ;

```

```

634 SORT      fcc      EXPLOS1-SPRITE ; #0 explosion start
635          fcc      EXPLOS2-SPRITE ; #1 explosion middle
636          fcc      EXPLOS3-SPRITE ; #2 explosion end
637 SORT2     fcc      PLANE-SPRITE2  ; #3 plane
638          fcc      LGEAR-SPRITE2  ; #4 landing gear
639          fcc      BOMBS-SPRITE2  ; #5 landing gear
640          fcc      EXPLOS-SPRITE2  ; #6 explosion bomb
641 ;-----
642 ; Variable section
643 ;-----
644 MHEIGHT fcc      5,7,9,10,0
645 HISCORE fcc      0,5
646          fcc      0,4
647          fcc      0,3
648          fcc      0,2
649          fcc      0,1
650 NAMES     fcc      'DEMO GAME DOS65',0
651          fcc      ' V.1.88 ',0
652          fcc      ' GAME LIBRARY ',0
653          fcc      ' (c) 1988 ',0
654          fcc      'DE 6502 KENNERS',0
655 ;
656 ; The KEYCODE and KEYSUB tables define the control keys and their subroutines
657 ; used in INPUT.MAC
658 ;
659 KEYCODE equ      $
660 FIRE     fcc      ' ' ;default FIRE key
661 GEAR      fcc      'g' ;default GEAR key
662 LAND      fcc      'l' ;default LAND key
663 QUIT      fcc      ESC ;default QUIT key
664 HALTKEY   fcc      'p' ;default HALT key
665 KEYEND    equ      $
666 KEYSUB    fdb      DROPB ;routine for FIRE key
667          fdb      MGEAR ;routine for GEAR key
668          fdb      LANDIT ;routine for LAND key
669          fdb      QADDR ;routine for QUIT key (see file KEY.MAC)
670          fdb      HADDR ;routine for HALT key (see file KEY.MAC)
671 NUMKEYS   equ      KEYEND-KEYCODE ;number of keys
672
673
674 BDIST     fcc      0 ;bomb distance
675 PLANEX    fcc      0 ;X position plane
676 PLANEY    fcc      0 ;Y position plane
677 GEAROUT    fcc      0 ;landing gear flag
678 LANDF     fcc      0 ;landing flag
679 BOMBX     res      10,0 ;X,Y positions bombs
680 BOMBY     res      10,0
681 SURFACE   res      80,0 ;store surface height
682
683          opt      nolis
684
685          lib      endgame ;end of game routines
686          lib      key ;key routines
687          lib      score ;score routines
688          lib      screen ;some small screen routines
689          lib      random ;random number generator
690          lib      sysclk ;get system clock utility for WAIT
691          lib      wait ;wait loop
692          lib      sprite ;plot/erase sprite
693
694          opt      lis
695          end      START

```


DE 6502 KENNER

MS-DOS Corner

Van Micro Ade naar MASM

Gert van Opbroek

Ik ben zelf geen MS-DOS man en toch schrijf ik nu onder de rubriek MS-DOS Corner, hoe is dat toch mogelijk? Mijn kennis van de compatibles en MS-DOS is zeer beperkt. Ik heb wel eens zo'n machine aangeraakt en ik het zelfs wel eens een commando zoals DIR ingetikt maar daar houdt het wel mee op. Waarom dan toch dit artikelje? Om de plannen die ik met de 6502 Kenner heb duidelijk te maken.

Wat is er aan de hand? Binnen het bestuur zijn we tot de conclusie gekomen dat onze club niet nog vele jaren van alleen de 6502 kan bestaan. Daarom is nu het moment aangekomen dat we alle types processoren binnen de club welkom heten. De achtergronden hebben we in nummer 57 uitgelegd. Deze wijziging wordt op de ledenvergadering in Almelo aan de leden voorgelegd.

Als redactie lopen we al vast vooruit op de hopelijk gunstige beslissing van de leden. Ik denk dat het goed is, alvast ruimte in de 6502 Kenner in te ruimen voor MS-DOS, bij deze dus. Veder denk ik dat het goed is uit te leggen welke kant de redactie uit wil gaan met de MS-DOS Corner.

De wortels van de club liggen in de industrie. De club is opgericht door een aantal enthousiaste mensen die bij hun werkgever de KLM hadden leren kennen. Om de kennis over dit systeem verder te verspreiden, is de Kim Gebruikersclub Nederland opgericht. Door de jaren heen, hebben de leden zich vooral bezig gehouden met zelfbouw van systemen (Junior, EC-65, DOS-65), met extra hardware (Eprommer) en met het programmeren in assembler. De laatste tijd zijn daar het programmeren in hogere programmeertalen (Basic, Forth en C) en de algemene artikelen over computerkunde bijgekomen. Welnu al deze zaken kunnen we ook voor andere systemen dan voor bijvoorbeeld DOS-65 gaan doen. Ik denk dat, vooral voor MS-DOS, hiervoor mogelijkheden te over zijn.

Zelfbouw

Het zelfbouwen van een PC is weliswaar mogelijk maar ik denk niet dat veel leden hier het geld voor over hebben. Voor mensen die hier wel oren naar hebben verwijs ik naar het project van het Duitse blad MC waar men een complete, modulaire AT rond een 80286 of 80386 heeft opgebouwd. Dit systeem kan men als bouw pakket kopen.

Inlichtingen hierover kan men bij de redactie krijgen.

Wat waarschijnlijk wel aantrekkelijk is, zijn zelfbouw-uitbreidingskaarten. Ook hiervoor zijn er in MC al voorbeelden te vinden, bijvoorbeeld in het oktobernummer een kaart met 32 input- en 32 outputlijnen verder bijvoorbeeld ADC- en DAC-kaarten, eprommers of epromkaarten. Ik heb zelfs al opgevangen dat er mogelijkheden zijn een elektuurbus te koppelen aan de uitbreidings slots van een PC. Indien er interesse bestaat, laat het ons even weten.

Wat ook wel gedaan wordt, is de PC gebruiken als ontwikkelmachine voor single board computers. Men gaat soms zelfs zo ver dat deze single board computer een transputer bevat. U ziet dus, mogelijkheden te over.

Programmering.

Zoals in een vorige paragraaf al aangehaald is, zijn we als club begonnen met het programmeren in assembler. Waarom zouden we dat niet ook voor een PC doen? Er bestaan assemblers voor de PC en waarom zullen we ons als club daar niet uitgebreid mee bezig gaan houden? Ook hogere programmeertalen zijn interessant waarbij ik vooral denk aan het zogenaamde machine-nabij programmeren. Ik weet wel dat er voor de PC al een hoop bestaat, maar vaak wil je toch net even iets anders als anderen.

Kennis.

Verder kunnen we ons als club bezig gaan houden met de uitwisseling van basiskennis over de systemen. Zelf heb ik door mijn opleiding en interesses een brede belangstelling voor alles wat met computers te maken heeft. Ik tracht de kennis die ik opgedaan heb een beetje te verspreiden door het schrijven van artikelletjes over algemene zaken, anderen kunnen misschien een bijdrage leveren door over specifiek de 8088 of MS-DOS te schrijven. Ik zou graag eens in de 6502 Kenner willen lezen hoe MS-DOS nu werkelijk in elkaar zit en wat de functie van de BIOS is. Ik heb met Nico de Vries afgesproken dat hij de spits af zal bijten met een artikel over de 8088.

U ziet, er zijn genoeg mogelijkheden, ik zoek nu alleen nog een redacteur voor de MS-DOS Corner. Misschien iets voor U?

NB. Ik zal nooit meer, zoals in nr. 57, schrijven dat de PC van de Z80 afstamt. Uit deze misser kunt u aflezen dat ik echt een PC-onbenul ben.

TECHNITRON TLP-12 LASER PRINTER

— U HEEFT EIGENLIJK GEEN ANDERE KEUZE!



- 12 pagina's per minuut (max.)
- tot 10.000 afdrukken per maand
- 8 ingebouwde lettertypes;
32 afdruk-combinaties
- unieke "FontMaker" service
- unieke "FormsMaker",
formulier- en logo service
- 3 ingebouwde hardware-
emulaties
- flexibele in- en uitvoer van papier

Technitron
DATA

Technitron Data B.V.
Zwarteweg 110, Postbus 14,
1430 AA Aalsmeer
tel. 02977-22456
telefax 02977-40968
telex 13301

Vestigingen in:

BONDSREPUBLIEK DUITSLAND – DENEMARKEN – ENGELAND – FRANKRIJK – ITALIË – NOORWEGEN – VERENIGDE STATEN – ZWEDEN